When Parametric Knowledge Wins: A Controlled Ablation of Agent Skills and Tool Use for PII Detection in Small Language Models

Anonymous authors
Paper under double-blind review

Abstract

Agent augmentation is widely assumed to improve performance, yet this study presents that for small language models, it systematically degrades capability under controlled conditions. This paper identifies a structural failure mode in agentic pipelines: agent augmentation that is assumed to benefit capable models systematically degrades performance in the 7–9B parameter class. A controlled ablation was run across four open-weight instruction-tuned models (Gemma 2 9B, Llama 3.1 8B, Mistral 7B, Qwen 2.5 7B) and four conditions: zero-shot prompting, documentation injection (+Docs), tool access (+Tool), and skills injection (+Skills). The benchmark is a stratified 2,000-sample dataset drawn from three public PII sources and scored against PII-Codex canonical types after full label alignment. A systematic capability regression is presented caused by agent augmentation in 7–9B parameter models. Under strict canonical-to-canonical scoring, zero-shot prompting outperforms every augmented condition for every included model in the 7–9B class. Tool use and skills injection reduce mean F1 by 13 to 24 percentage points relative to zero-shot ($p < 0.0001$, Cohen's d from -0.39 to -0.67). Documentation is mostly neutral, though it significantly hurts Llama 3.1 8B ($\Delta = -0.17$). Adding a Skill document on top of tool access provided no measurable benefit for any model. The degradation is not uniform. Structured types like Date and IP Address actually improve under tool use, while temporal (Date Time) and medical (Health Insurance ID) types collapse near zero, driven by label-schema mismatches between PII-Codex output and ground truth. Implications are discussed for evaluation methodology and agentic pipeline design in the 7–9B parameter class.

1 Introduction

The proliferation of agentic AI systems has introduced a new class of prompting strategies in which language models are given access to external tools, domain documentation, and structured skill definitions at inference time (Yao et al., 2023; Schick et al., 2023; Chase, 2022). The motivating intuition is straightforward: if a model lacks reliable internal representations of a specialized domain, injecting authoritative external references should improve task performance. This assumption underpins a broad class of retrieval-augmented generation (RAG) systems (Lewis et al., 2020), tool-use frameworks, and the emerging concept of *Agent Skills*—modular, reusable context documents that encode procedural knowledge, tool syntax, and domain constraints for a specific task.

PII detection is a natural testbed for this hypothesis. It is a high-stakes structured extraction task with a well-defined label schema, and it is increasingly delegated to SLMs in agentic data pipelines where latency, cost, and privacy constraints preclude the use of frontier-scale models (Lison et al., 2021; Lukas et al., 2023). Sending sensitive text to API-based frontier models pose unacceptable privacy and security risks; deploying local SLMs for PII and data-science workloads is therefore a critical industry requirement (Sun et al., 2025). The domain also has a mature programmatic library ecosystem including Microsoft Presidio (Mendels et al., 2018), spaCy NER (Honnibal et al., 2020), PII-Codex (Rosado, 2023), which provide structured canonical

taxonomies against which model predictions can be systematically compared, though programmatic detection accuracy varies by entity type.

A four-condition ablation was designed to isolate the effects of documentation, tool access, and skills injection on PII detection accuracy in four open-weight SLMs (7–9B parameters), including Gemma 2 9b (Team et al., 2024), Llama 3.1 8b (Grattafiori et al., 2024), Mistral 7b (Jiang et al., 2023), and Qwen 2.5 7b (Yang et al., 2024). The conditions are:

1. **Zero-shot (+ZS):** The model receives a task prompt and sample text with no external augmentation.
2. **Documentation (+Docs):** The model additionally receives the PII-Codex reference documentation in context (single-turn; no tool execution).
3. **Tool-augmented (+Tool):** The model is given access to the `analyze_pii` tool (PII-Codex execution) via a LangGraph-based SkillsAgent, and may call it before returning a final answer.
4. **Skills-augmented (+Skills):** In addition to tool access, the model may discover and read a structured PII detection Skill document (`SKILL.md`) via `list_skills` and `view_skill` tool calls before calling `analyze_pii`.

Using a stratified benchmark of 2,000 PII-annotated English samples from three public datasets, with post-hoc label alignment to PII-Codex canonical types, zero-shot prompting was found to outperform all augmented conditions for all four models, a result that inverts the common practitioner assumption. It aligns with the view that modern SLMs already possess strong inherent reasoning and text-understanding capabilities (Yu et al., 2024) and need not rely on external tools for a fundamental extraction task like PII detection, and with recent evidence that arbitrary tool access can actively degrade performance compared to zero-shot when retrieval errors and scaffolding overhead outweigh tool benefits (Luo et al., 2025; Qian et al., 2025; Xu et al., 2025). LLMs struggle to decide *whether* to use a tool (Ning et al., 2024); zero-shot prompting avoids that pitfall by not deferring to external tools when parametric knowledge suffices. The finding has direct implications for how agentic scaffolding should be evaluated and deployed in production NLP pipelines.

2 Related Work

PII detection with language models. Named entity recognition (NER) approaches to PII detection have a long history in NLP (Lample et al., 2016; Devlin et al., 2019). More recently, instruction-tuned LLMs have been evaluated as zero-shot or few-shot entity extractors, though their out-of-the-box performance on strict sequence-labeling tasks often lags behind supervised baselines without highly specialized prompt adaptations (Wang et al., 2025). Furthermore, separate work demonstrates that language models are highly susceptible to leaking sensitive PII depending on the prompt design and the entity type targeted by an attacker, underscoring the critical need for reliable, external PII detection safeguards in agentic pipelines (Lukas et al., 2023). PII-Codex (Rosado, 2023) is built on Microsoft Presidio NER detection (Mendels et al., 2018), leverages the information-sensitivity typology (Milne et al., 2016) and the risk identification continuum (Schwartz & Solove, 2011); it provides a unified taxonomy and programmatic evaluation framework that enables type-level scoring against a canonical schema.

Tool use and RAG in language models. Surveys of natural language reasoning suggest that modern SLMs already possess strong inherent reasoning and text-understanding capabilities (Yu et al., 2024); for a static, pattern-recognition task like PII detection, external tools may therefore be redundant or harmful. Conversely, tools are necessary when models must incorporate real-time, rapidly changing, or newly updated information that cannot be captured by static pretrained weights (Yu & Ji, 2024); PII detection, by contrast, is a static extraction task for which parametric knowledge and in-context instruction suffice, and forcing an agent to invoke tools introduces unnecessary overhead and the degradation observed in this study. ReAct (Yao et al., 2023) and Toolformer (Schick et al., 2023) established that language models can interleave reasoning and tool execution. Subsequent work has shown that tool use can substantially improve performance

on knowledge-intensive tasks, though gains are not universal and depend heavily on instruction-following fidelity and the schema alignment between tool outputs and the evaluation metric (Qin et al., 2024). Providing models with arbitrary tool access can actively degrade performance compared to zero-shot, as error rates from unnecessary retrieval, inappropriate invocation, and scaffolding often outweigh tool benefits (Luo et al., 2025; Qian et al., 2025; Xu et al., 2025). Empirically, LLM-generated internal knowledge can surpass externally retrieved knowledge; external retrieval often introduces irrelevant or poorly coherent information that harms downstream performance more than minor factual errors in parametric memory (Chen et al., 2023). For many tasks, parametric knowledge is sufficient and "tool overuse" lowers accuracy (Luo et al., 2025; Qian et al., 2025). SLMs generally fail to execute multi-turn tool-discovery loops out-of-the-box; targeted domain-specific fine-tuning is often advocated for reliable agentic tool calling (Belcak et al., 2025; Jhandi et al., 2025). Toolformer showed that a small model can excel at tool use when fine-tuned to predict tool calls (Schick et al., 2023); out-of-the-box scaffolding without such fine-tuning is a poor fit for the 7–9B class (Patil et al., 2024). The study contributes an evaluation of tool use in the specific context of structured information extraction in the 7–9B parameter class.

Agent Skills and skill injection. The concept of injecting modular skill documents into model context at inference time is a practitioner pattern with limited formal evaluation in the literature.

This study provides, to the best of the author's knowledge, the first controlled ablation of skills injection for PII detection, with pre-registered conditions and post-hoc statistical testing.

3 Methods

3.1 Benchmark compilation

A stratified benchmark of 2,000 English-language PII-annotated samples was compiled from three public HuggingFace datasets, AI4Privacy (Ai4Privacy, 2024) (`ai4privacy/pii-masking-300k`), NVIDIA Nemotron-PII (Steier et al., 2025) (`nvidia/Nemotron-PII`), and Gretel PII Masking (AI, 2024) (`gretelai/gretel-pii-masking-en-v1`).

Let $D = \{(x_i, G_i)\}_{i=1}^{N}$ with $N = 2{,}000$ denote the benchmark, where x_i is a text sample and $G_i = \{(s_{ij}, y_{ij})\}$ is the set of ground-truth entity spans and types for sample i. Source distributions in the final benchmark are shown in Table 1. The mix of PII types varies by source; type-level counts are in Appendix Table 6.

Table 1: Benchmark source distribution (N = 2,000). Samples and per-source entity counts as recorded during stratification. Total entity span count for the compiled benchmark is given in Appendix Table 6.

Source	Samples	Entity spans (source)	Mean/sample
AI4Privacy	1,132	7,112	6.28
Gretel PII Masking	581	2,416	4.16
NVIDIA Nemotron-PII	287	1,842	6.42
Total	**2,000**	**11,370**	**5.69**

Language and locale filtering. AI4Privacy was filtered to rows with `language == "English"`; NVIDIA Nemotron-PII to `locale == "us"`; Gretel requires no filter (English-only dataset). Results apply to English text and US-locale PII types only and do not generalize to other locales or languages.

Label mapping and exclusion. Source labels were normalized and mapped to PII-Codex (Rosado, 2023) canonical types (`PIIType.name`) via a fixed 136-entry mapping table. Records containing any PII instance with no PII-Codex mapping were excluded, so all ground-truth labels in the benchmark are fully mappable. The final set covers 21 PII types. Ground-truth entity span counts (each contiguous span of a given type

counted once) by type for the full benchmark are in Appendix Table 6; the most frequent types are `PERSON`, `DATE_TIME`, `LOCATION`, `ADDRESS`, and `EMAIL_ADDRESS` (Table 2).

Table 2: Selected frequent PII types by ground-truth entity count (main study benchmark, descending). Full distribution in Appendix Table 6.

PII Type	Entity count
PERSON	2,324
DATE_TIME	1,572
LOCATION	1,068
ADDRESS	862
EMAIL_ADDRESS	632
US_SOCIAL_SECURITY_NUMBER	632
PHONE_NUMBER	566
DATE	531
IP_ADDRESS	515
US_DRIVERS_LICENSE_NUMBER	383
HEALTH_INSURANCE_ID	343

3.2 Models

Four open-weight instruction-tuned models were evaluated: Gemma 2 9B, Llama 3.1 8B, Mistral 7B, and Qwen 2.5 7B. All models were run using 4-bit quantized MLX variants (`mlx-community`) on a 2020 MacBook Pro M1 Pro Max (64 GB RAM). Prompts and chat templates were identical across models and conditions; seed 42 was fixed for all runs.

3.3 Conditions

Each of the 2,000 benchmark samples was evaluated under four conditions, yielding 32,000 total prediction rows (2,000 samples $\times$ 4 models $\times$ 4 conditions). Each condition c defines an inference pipeline $f_c \colon \mathcal{X} \to \mathcal{P}(\mathcal{S} \times \mathcal{Y}_c)$ mapping input text x_i to a predicted set of entity spans and labels $\hat{G}_{i,c} = f_c(x_i)$.

Zero-shot (ZS). Single-turn prompt containing the task instruction and sample text. No external augmentation. The model returns a JSON array of PII predictions.

Documentation (+Docs). Same single-turn path with the PII-Codex reference documentation prepended to the prompt. No tool execution.

Tool-augmented (+Tool). The model is given access to `analyze_pii`, which executes PII-Codex on the sample text, via a custom SkillsAgent created with LangGraph (LangChain, 2024; Chase, 2022). Tool-call detection uses strict bracket syntax (`[TOOL_CALL: analyze_pii]`) with a lenient intent-based fallback. If no tool call is detected, the model's first response is taken as the final answer.

Skills-augmented (+Skills). Same LangGraph (LangChain, 2024) agent with two additional tools: `list_skills` (returns available skill names) and `view_skill("pii-detection")` (returns the PII detection Skill document, `SKILL.md`). Models that read the Skill document before calling `analyze_pii` receive a structured prompt encoding PII-Codex tool syntax, output format, and type coverage. The same Skill document was used for both pilot and main studies.

3.4 Execution infrastructure

Conditions were run sequentially per model (one condition at a time). Within multi-turn conditions (+Tool, +Skills), samples were processed in parallel via a ThreadPoolExecutor (1 worker in this study), with a

shared lock serializing `generate` calls. Context was bounded per turn (initial user prompt + current system message only) to prevent context growth from degrading performance in the skills discovery loop. A per-sample timeout prevented runaway inference without discarding partial results.

3.5 Scoring and label alignment

Predictions were scored against ground truth using precision, recall, and macro-averaged F1 at the entity-type level, with span IoU used where character offsets were available. Due to zero-shot and documentation conditions producing primarily model-native label names (e.g., `person_name`, `ssn`, `date of birth`) that differ from PII-Codex canonical types, all prediction labels were normalized and mapped to PII-Codex types via a 136-entry mapping table (`pii_label_to_piicodex.json`) before scoring. A fixed alignment map $\phi_c \colon \mathcal{Y}_c \to \mathcal{Y}_{\text{canon}}$ maps condition-specific label vocabularies to the canonical PII-Codex schema; aligned predictions are $\tilde{G}_{i,c} = \{(s, \phi_c(y)) \mid (s, y) \in \hat{G}_{i,c}\}$. Pre-alignment scores are not comparable across conditions and are not reported; all results below are *post-alignment*.

This label alignment step is critical for valid cross-condition comparison: tool and skills conditions output PII-Codex canonical types directly (because PII-Codex is the tool being called), while zero-shot and documentation conditions use model-native labels that require remapping. Without alignment, zero-shot appears artificially low, and tool conditions appear artificially high.

3.6 Statistical analysis

Each of the 2,000 benchmark samples appears in all four conditions, forming a within-subjects design. Condition effects are estimated from paired per-sample deltas $\Delta_i(c) = \text{F1}(\tilde{G}_{i,c}, G_i) - \text{F1}(\tilde{G}_{i,\text{ZS}}, G_i)$, constituting a repeated-measures design. Reported are paired t-tests (condition F1 vs. zero-shot F1 at the sample level) and Cohen's d for key comparisons (+Docs vs. ZS, +Tool vs. ZS, +Skills vs. ZS, and +Skills vs. +Tool). Bootstrap 95% confidence intervals are reported for mean F1 per (model, condition) cell. Sample size was confirmed sufficient for the target margin of ± 0.05 in all 16 cells (maximum required $n = 177$; actual $n = 2,000$).

4 Results

4.1 Aggregate performance

Table 3 presents post-alignment mean F1 by model and condition. Under strict canonical-to-canonical scoring, zero-shot achieves the highest F1 for all four models. Tool and skills conditions underperform zero-shot by 13–24 percentage points. Documentation is largely neutral (Gemma, Mistral, Qwen: $\Delta \approx 0$) but substantially harmful for Llama 3.1 8B ($\Delta = -0.17$). Figure 1 shows the same comparison with bootstrap 95% intervals.

Table 3: Main study mean F1 by model and condition (N = 2,000, post-label alignment). All deltas are relative to zero-shot.

Model	ZS	+Docs	Δ_{Docs}	+Tool	Δ_{Tool}	+Skills	Δ_{Skills}
Gemma 2 9B	0.56	0.56	0.00	0.40	−0.16	0.40	−0.16
Llama 3.1 8B	0.62	0.45	−0.17	0.44	−0.18	0.38	−0.24
Mistral 7B	0.54	0.53	−0.01	0.41	−0.13	0.41	−0.13
Qwen 2.5 7B	0.57	0.56	−0.01	0.37	−0.20	0.38	−0.19

4.2 Statistical significance and effect sizes

Table 4 reports paired t-test results and Cohen's d for each model and condition comparison against zero-shot. All +Tool and +Skills comparisons (vs. zero-shot) are statistically significant at $p < 0.001$ with medium-to-

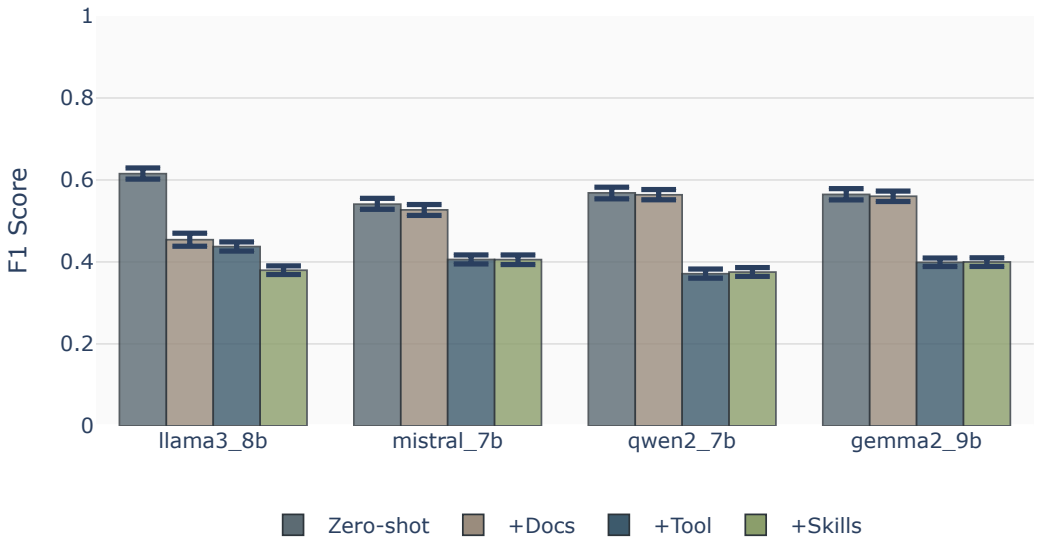

Figure 1: Mean F1 by condition and model (main study, n = 2,000, post-label alignment). Bars show bootstrap 95% CI. Zero-shot (ZS) exceeds +Tool and +Skills for all models; +Docs is close to ZS for three of four models.

large negative effect sizes. Documentation (+Docs vs. zero-shot) has no significant effect for Qwen or Gemma ($p = 0.54$ and $p = 0.53$ respectively); Mistral's documentation effect is borderline ($p = 0.048$, $d = -0.04$); Llama's is statistically significant ($p < 0.0001$, $d = -0.40$).

The incremental effect of adding the Skill document on top of tool access (+Skills vs. +Tool) yields Cohen's d of -0.29 for Llama, -0.01 for Mistral, $+0.02$ for Qwen, and $+0.05$ for Gemma. None reach conventional thresholds for a meaningful effect, confirming that the Skill document provides no consistent incremental benefit beyond tool access alone. Figure 2 illustrates the distribution of per-sample deltas.

4.3 Per-PII-type recall

Aggregate F1 masks heterogeneous effects across PII types. High-support types like `PERSON` and `DATE_TIME` contribute disproportionately to aggregate F1, so degradation in those types drives the overall decline observed across conditions. Table 5 shows mean recall delta (+Skills vs. zero-shot) for the ten most frequent types; Figure 3 summarizes recall by PII type and condition across models, and Figure 4 gives the full recall-by-type view by model. Full recall-by-type values for all four conditions and models are in Appendix E (Table 11).

Types that improve: Structured identifiers with unambiguous canonical forms (`DATE`, `US_SOCIAL_SECURITY_NUMBER`, `IP_ADDRESS`) consistently gain recall under tool use and skills injection across all models. This is consistent with PII-Codex having reliable detection rules for these types.

Table 4: Paired t-test results and Cohen's d (condition vs. zero-shot, N = 2,000 per model).

Model	Condition	t	p	d
Llama 3.1 8B	+Docs	−17.94	< 0.0001	−0.40
	+Tool	−23.17	< 0.0001	−0.52
	+Skills	−29.94	< 0.0001	−0.67
Mistral 7B	+Docs	−1.98	0.048	−0.04
	+Tool	−17.59	< 0.0001	−0.39
	+Skills	−17.63	< 0.0001	−0.39
Qwen 2.5 7B	+Docs	−0.61	0.543	−0.01
	+Tool	−24.45	< 0.0001	−0.55
	+Skills	−23.81	< 0.0001	−0.53
Gemma 2 9B	+Docs	−0.63	0.530	−0.01
	+Tool	−20.99	< 0.0001	−0.47
	+Skills	−20.88	< 0.0001	−0.47

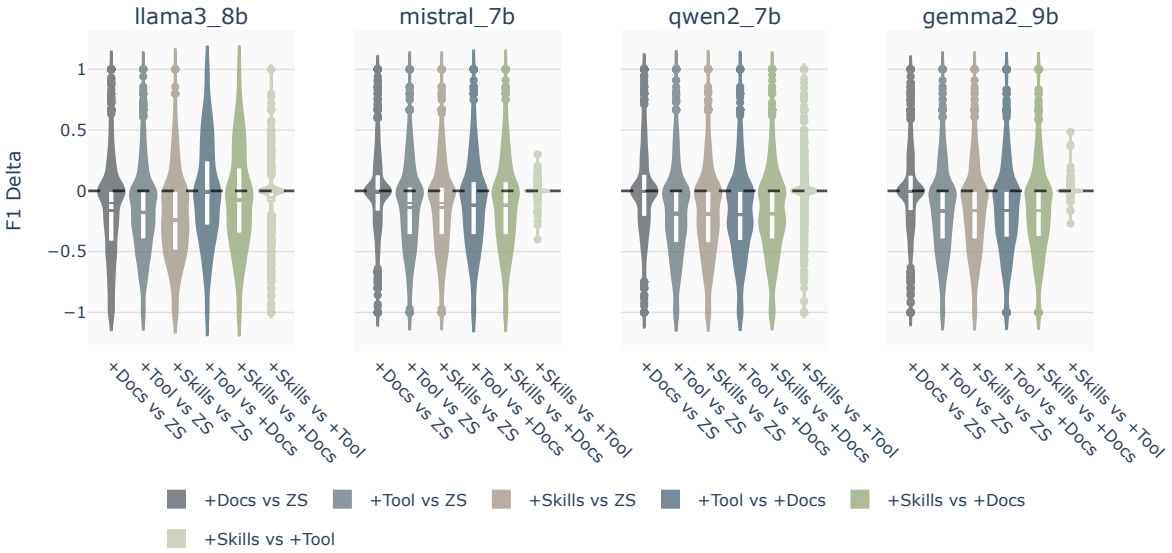

Figure 2: Distribution of per-sample F1 deltas (condition minus zero-shot) by model. Negative values indicate that the condition underperforms zero-shot; +Tool and +Skills show consistent negative shifts.

Types that degrade: Temporal (`DATE_TIME`) and medical (`HEALTH_INSURANCE_ID`) types collapse nearly to zero under all tool conditions. This was attributed to a label-schema mismatch: in tool and skills conditions, models write `DATE` in their final prediction JSON for temporal spans, and the label alignment map does not bridge `DATE` to `DATE_TIME`, so the benchmark's `DATE_TIME` entities go unmatched at scoring time. Similarly,

Table 5: Recall delta (+Skills vs. zero-shot) by PII type and model (main study). Positive values indicate improvement under skills injection; negative values indicate degradation.

PII Type	Gemma 2 9B	Llama 3.1 8B	Mistral 7B	Qwen 2.5 7B
DATE	**+0.64**	**+0.24**	**+0.52**	**+0.18**
US_SOCIAL_SECURITY_NUMBER	**+0.26**	**+0.28**	**+0.01**	**+0.38**
IP_ADDRESS	**+0.27**	**+0.09**	**+0.20**	**+0.05**
EMAIL_ADDRESS	**+0.02**	**+0.02**	**+0.04**	**+0.00**
ADDRESS	−0.02	−0.18	−0.01	−0.16
PERSON	−0.20	−0.25	−0.05	−0.17
PHONE_NUMBER	−0.12	−0.13	−0.18	−0.29
LOCATION	−0.21	−0.36	−0.15	−0.44
DATE_TIME	−0.26	−0.52	−0.29	−0.33
HEALTH_INSURANCE_ID	−0.54	−0.64	−0.53	−0.57

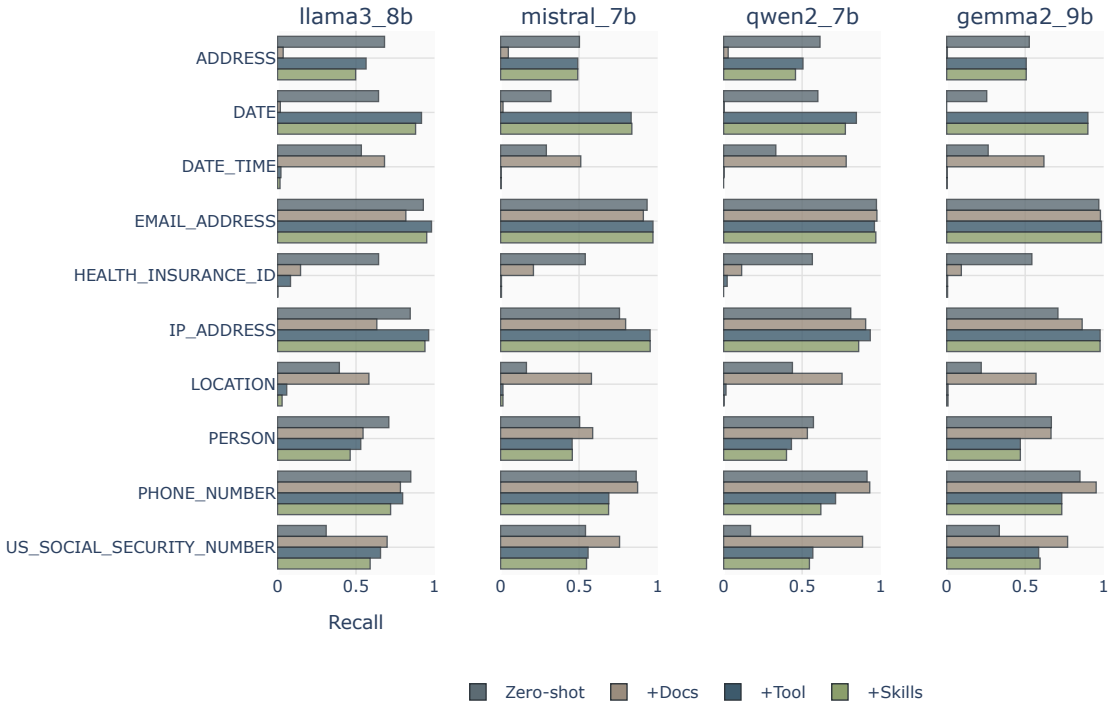

Figure 3: Recall by PII type and condition (main study). Summarizes how each model and condition performs across the benchmark's top 10 PII types.

`HEALTH_INSURANCE_ID` recall drops to near zero under tool use across all models, suggesting that PII-Codex does not reliably detect this type at the entity level.

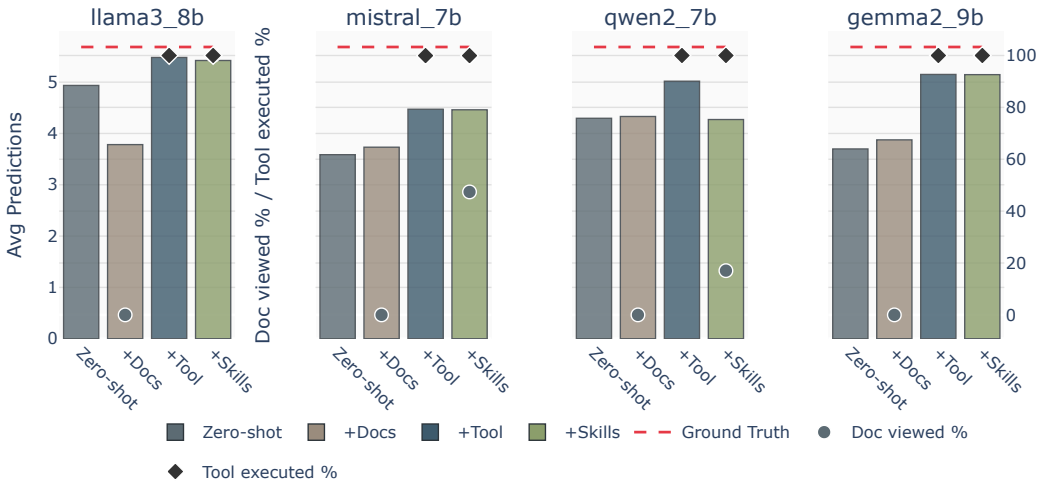

Figure 4: Mean recall by PII type and condition, faceted by model (main study). Structured types (e.g., DATE, SSN, IP) often gain under +Tool/+Skills; DATE_TIME and HEALTH_INSURANCE_ID recall drops under tool conditions.

4.4 Skill-viewed subgroup analysis

For the +Skills condition, Mistral read the Skill document in 948 of 2,000 runs (47.4%) and Qwen in 341 of 2,000 runs (17.1%). Llama and Gemma read the document in 100% of runs (no non-viewed comparison available). For Mistral (viewed $n = 948$, not-viewed $n = 1,052$) and Qwen (viewed $n = 341$, not-viewed $n = 1,659$), mean F1 was visually indistinguishable between the two groups in both cases. This within-condition comparison controls for all factors except document reading, and provides no evidence that reading the Skill document improves PII detection performance, even when the model chooses to access it.

4.5 Execution reliability

Tool execution was reliable across both tool conditions: the 4,000 executions per model (2,000 samples $\times$ 2 conditions: +Tool and +Skills) completed successfully for Gemma, Llama, and Qwen; Mistral completed 3,998/4,000 (99.9%). Error rate was 0% across all 16 model-condition cells in both pilot and main study runs. The F1 degradation under tool conditions is therefore attributable to the quality of tool output and label schema alignment, not to execution failures.

5 Discussion

5.1 The pre-alignment confound

The central finding is not methodological but behavioral: agent augmentation introduces systematic interference that reduces performance in small models. Evaluation confounds explain the mechanism, not the phenomenon. This is because tool conditions invoke PII-Codex directly and therefore output canonical types that score without remapping, while zero-shot outputs require a post-hoc alignment step. A study that reports results before label alignment would incorrectly conclude that tool use substantially improves PII detection. After alignment, the comparison is fair, and the direction of effect reverses. The parallel is sharp: just as superficial metrics in model editing can mask underlying failures and yield systematically

misleading conclusions about a technique's success (Baser et al., 2026), comparing raw tool versus zero-shot scores without normalizing the label schema gives a false sense of tool superiority.

This confound is not specific to this study's design; it is a general risk for any evaluation framework that mixes model-native and tool-canonicalized predictions without explicit normalization. The author recommends that future benchmarking of agentic PII systems include a mandatory label alignment pass and report pre- and post-alignment scores separately.

5.2 Why does tool use degrade aggregate F1?

The per-type analysis in Section 4.3 suggests two mechanisms. First, in tool and skills conditions, models consistently write `DATE` in their final prediction JSON for temporal spans (15,956 `DATE` predictions vs. 21 `DATE_TIME` across 78,847 prediction entities in those conditions). Because the label alignment map does not bridge `DATE` to `DATE_TIME`, and the benchmark contains 1,572 `DATE_TIME` entities, these predictions go unmatched at scoring time, collapsing `DATE_TIME` recall to near zero.

Finding. If no aligned prediction has type k, i.e., $\forall(s, y) \in \hat{G}_{i,c}$, $\phi_c(y) \neq k$, then $\mathrm{TP}_k^{(c)} = 0$ and thus $\mathrm{Recall}_k^{(c)} = 0$. Under the scoring rule, a match requires aligned type equality; no prediction then contributes to true positives for type k. Documentation increases `DATE_TIME` recall in some models (e.g., Llama), suggesting that label vocabulary priming alone does not cause the collapse; the collapse emerges specifically when tool canonicalization is invoked.

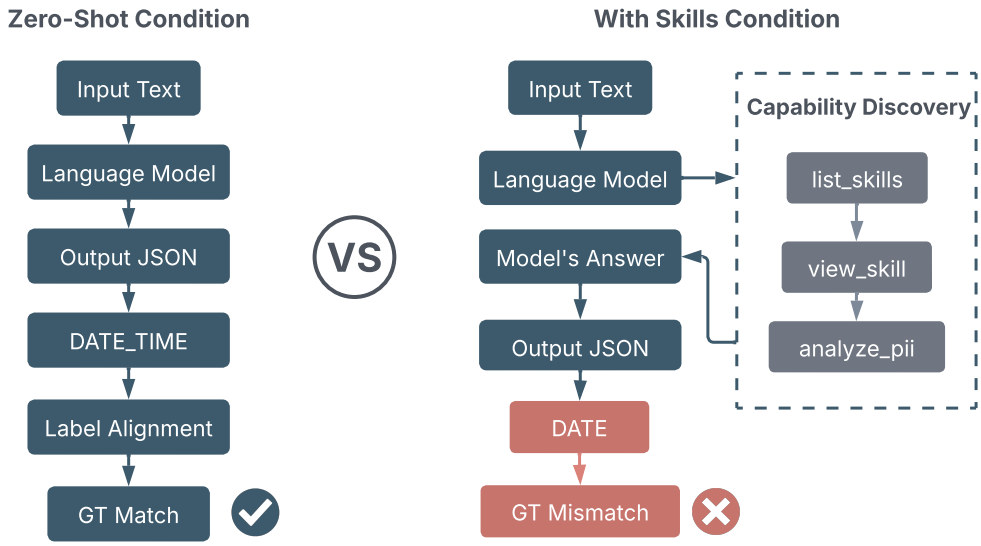

Figure 5: The labeling confound: zero-shot predictions are aligned to the evaluation schema before scoring; tool-augmented outputs are already canonical (e.g., PII-Codex `DATE`) but may not match ground-truth types (e.g., `DATE_TIME`), so aligned type equality fails and recall collapses.

Second, for types like `HEALTH_INSURANCE_ID` and `LOCATION`, PII-Codex appears to have lower recall than the models themselves under zero-shot prompting. This suggests that model pretraining knowledge of PII categories may outperform a rule-based programmatic library for contextually ambiguous types, even when the model is operating without any augmentation (Yu et al., 2023; Chen et al., 2023; Luo et al., 2025; Qian et al., 2025). As in knowledge-intensive tasks where generated internal knowledge surpasses externally retrieved knowledge (Yu et al., 2023), external programmatic tools for PII lack the nuanced contextual understanding that neural representations possess; the coherence of model-native output can outweigh the occasional benefit of rule-based canonicalization. The contrast aligns with the distinction between settings

where tools are required—e.g., when models must integrate real-time or frequently updated information that exceeds their pretrained knowledge (Yu & Ji, 2024)—and settings where tools are superfluous: PII detection is a static, pattern-recognition task, and injecting tool-based scaffolding introduces structural mismatch and the degradation observed here. Models often lack reliable self-awareness of their knowledge boundaries (Yin et al., 2023), which may explain why they eagerly invoke tools even when tool output underperforms their own parametric knowledge. The observed degradation may reflect a representational mismatch between neural contextual extraction and rule-based canonical detectors.

The `DATE`/`DATE_TIME` recall collapse is therefore partly a scoring design choice: the alignment map does not treat `DATE` as a parent of `DATE_TIME`. An alternate scoring scheme could assign partial or full credit when a tool outputs `DATE` for ground-truth `DATE_TIME` (hierarchical or relaxed temporal mapping), or report span-only (type-agnostic) F1 alongside type-level F1 to separate "finding the span" from "labeling it." Reporting both strict canonical-to-canonical scores and a relaxed temporal mapping would strengthen the claim that the observed degradation is pipeline- and evaluation-driven rather than evidence that the tool is strictly worse at finding temporal spans.

5.3 Documentation and the Llama exception

Three of four models show near-zero sensitivity to documentation injection ($|\Delta| \leq 0.01$). Llama 3.1 8B is a notable outlier, with a statistically significant -0.17 degradation under +Docs ($p < 0.0001$, $d = -0.40$). This is consistent with recent findings that a model's stronger baseline instruction-following ability can paradoxically degrade performance by making it overly sensitive to rigid or suboptimal formatting constraints in the prompt (Luo et al., 2025). Furthermore, injecting external documentation can create "merge conflicts" with a model's parametric knowledge (Qian et al., 2023). When given the PII-Codex documentation, Llama may attempt to strictly map its output to the documentation's label set rather than relying on its own more accurate internal representations.

5.4 Skill document reading does not explain performance

The skill-viewed subgroup analysis (Section 4.4) provides strong evidence against the hypothesis that the Skill document has any incremental value. Models that read the document do not outperform models that do not, within the same condition. The document's failure to help may reflect a context competition effect: the Skill document consumes context capacity that could otherwise be used for the model's own chain-of-thought or attention to the sample text; dense instructions and tool outputs in the context window trigger context competition, degrading a model's ability to extract and use relevant information (Liu et al., 2024; Luo et al., 2025; Xu et al., 2025; Shi et al., 2023).

5.5 Implications for agentic NLP pipeline design

The results suggest a broader design principle: agent scaffolding imposes structural constraints that can conflict with small-model representations. Varying prompt phrasings or rigid output constraints lead to divergent implementations and metrics, undermining the reliability of agent evaluation (Sun et al., 2025).

1. **Do not assume tool use improves structured extraction.** Consistent with tool-use literature (Luo et al., 2025; Qian et al., 2025; Xu et al., 2025), tool-augmented pipelines in the 7–9B parameter class were found to add latency and infrastructure complexity while reducing accuracy relative to zero-shot prompting with label normalization. The benefit of tool use is type-specific (structured identifiers) rather than global.

2. **Label alignment is a first-class evaluation concern.** Any mixed-mode pipeline (some predictions from tool output, some from model output) must normalize labels before scoring. Omitting this step will produce systematically misleading comparisons.

3. **Model size and instruction-following fidelity moderate the effect.** The degradation from tool use is largest for models with the strongest zero-shot performance (Llama, Qwen), suggesting

that capability amplifies rather than buffers the cost of agentic scaffolding. This is consistent with evidence that explicitly constraining an LLM's answer format can actively harm reasoning and performance (Tam et al., 2024). Higher-capability models appear more disrupted because they more faithfully execute the rigid structural constraints imposed by the agent framework, creating a stronger conflict with their learned output representations than is observed in weaker models (Luo et al., 2025; Tam et al., 2024; Jhandi et al., 2025). Agentic scaffolding design cannot therefore treat model capability as a monotonically beneficial factor; the same instruction-following fidelity that drives strong zero-shot performance may be precisely what makes a model more susceptible to format-induced degradation in tool-augmented pipelines (Luo et al., 2025; Belcak et al., 2025), further exacerbating the fragility and reproducibility of agentic evaluations (Sun et al., 2025).

5.6 Limitations

This study focuses exclusively on English-language text and US-specific PII types. The findings, therefore, may not apply to other languages or international PII schemas. All models were assessed using 4-bit quantization, so results may vary with full-precision or larger-scale versions. The benchmark covers 21 PII types from three sources; coverage of rare types (e.g., `ABA_ROUTING_NUMBER`, `LICENSE_PLATE_NUMBER`) is limited. The study does not evaluate few-shot prompting, chain-of-thought prompting, or fine-tuned models, which represent important comparison points for future work.

Furthermore, the zero-shot advantage over tool-use observed in this study should be interpreted within the constraints of the experimental setup rather than as a general indictment of tool use for PII detection. Two design factors limit generalizability. First, the Skill document was deliberately compressed to approximately 150 tokens to prevent context overload in small models (Luo et al., 2025; Xu et al., 2025); a more richly specified document with worked examples and type disambiguation may produce different outcomes. Second, while massive, frontier-scale models can often adapt to tool-use via in-context prompting alone, foundational literature demonstrates that out-of-the-box models in the 7B to 9B parameter class generally fail to reliably execute rigid API calls without significant hallucinations (Patil et al., 2024; Qin et al., 2024). For instance, prior evaluations of base 7B models on zero-shot tool-calling benchmarks have yielded 0% accuracy and 100% hallucination rates, whereas the exact same architectures achieve state-of-the-art tool execution only after undergoing heavy, domain-specific fine-tuning on API datasets (Patil et al., 2024). Therefore, the degradation observed under the tool-augmented conditions reinforces the notion that applying complex agentic scaffolding to out-of-the-box 7–9B models is a structural mismatch; to achieve reliable tool execution at this scale, practitioners likely cannot rely on zero-shot prompting frameworks and must instead invest in targeted fine-tuning. The low skill-document engagement rates observed for Qwen (17.1%) and Mistral (47.4%) are consistent with that finding. These results are therefore best understood as specific to this model class, skill construction, and task configuration.

Finally, the LangGraph agent uses a lenient tool-call detection strategy; a stricter enforcement of tool use before accepting answers (available but not used in the default runner) may produce different results.

6 Conclusion

A controlled ablation study was presented evaluating the effect of documentation injection, tool access, and Agent Skills documents on PII detection accuracy across four open-weight SLMs. The primary finding is that, under strict canonical-to-canonical scoring, zero-shot prompting with post-hoc label alignment outperforms all augmentation conditions for all models in the 7–9B class, with tool and skills conditions reducing mean F1 by 13–24 percentage points relative to zero-shot. This reversal of the expected direction of effect is explained by a label-schema mismatch between PII-Codex tool output and the ground-truth benchmark, which disproportionately affects high-frequency temporal types. Structured identifier types (`DATE`, `US_SOCIAL_SECURITY_NUMBER`, `IP_ADDRESS`) do benefit from tool execution, suggesting that the value of programmatic PII tools is type-selective rather than global.

The skill-viewed subgroup analysis provides additional evidence that Skill document injection has no measurable incremental value, even for models that choose to read the document. Documentation injection is

neutral for most models but harmful for Llama 3.1 8B, suggesting model-specific sensitivity to context that conflicts with pre-trained label representations.

The primary contribution is identifying capability regression from agent augmentation in small models, with evaluation design explaining when and why the regression occurs. As in other domains, superficial evaluations obscure true capabilities: evaluation design, schema canonicalization, and ontology alignment fundamentally mediate how the community perceives model performance (Baser et al., 2026). Toolformer demonstrated that a 6.7B model can outperform larger models on tool use when *fine-tuned* to predict tool calls natively (Schick et al., 2023); zero-shot agentic scaffolding is therefore fundamentally mismatched for 7–9B models unless they are fine-tuned (Patil et al., 2024). These results challenge a common assumption in agentic system design and highlight the importance of rigorous label alignment in mixed-mode evaluation pipelines. Future work should examine whether these patterns hold for larger models, for few-shot and chain-of-thought prompting baselines, and for international PII types beyond the US-English scope of this study.

Broader Impact Statement

This work evaluates the effect of agent augmentation on PII detection accuracy in small language models. The primary practical implication is cautionary: practitioners deploying agentic scaffolding for sensitive data tasks in the 7–9B parameter class should validate augmentation decisions empirically rather than assuming performance gains. Uncritical adoption of tool-augmented pipelines for PII detection could reduce accuracy relative to simpler zero-shot approaches, with downstream consequences for privacy compliance and data protection in production systems.

The benchmark, scored results, and analysis notebooks are publicly released to support reproducibility and to enable the community to verify, extend, or challenge these findings. All datasets used are synthetic or publicly available; no real personal data was used or exposed in this research.

Data and Code Availability

Code and dataset made available to the public. Links are presently redacted due to double-blind review.

Acknowledgments

Omitted for anonymous review.

References

Gretel AI. Gliner models for pii detection through fine-tuning on gretel-generated synthetic documents, 10 2024.

Ai4Privacy. pii-masking-300k (revision 86db63b), 2024. URL `https://huggingface.co/datasets/ai4privacy/pii-masking-300k`.

Manit Baser, Dinil Mon Divakaran, and Mohan Gurusamy. Thinkeval: Practical evaluation of knowledge leakage in LLM editing using thought-based knowledge graphs. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small language models are the future of agentic ai, 2025. URL `https://arxiv.org/abs/2506.02153`.

Harrison Chase. Langchain, 2022. URL `https://github.com/langchain-ai/langchain`. Computer software.

Liang Chen, Yang Deng, Yatao Bian, Zeyu Qin, Bingzhe Wu, Tat-Seng Chua, and Kam-Fai Wong. Beyond factuality: A comprehensive evaluation of large language models as knowledge generators. In *Proceedings*

of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 6325–6341, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.390. URL https://aclanthology.org/2023.emnlp-main.390/.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim,

Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength natural language processing in python. 2020. doi: 10.5281/zenodo.1212303.

Polaris Jhandi, Owais Kazi, Shreyas Subramanian, and Neel Sendas. Small language models for efficient agentic tool calling: Outperforming large models with targeted fine-tuning, 2025. URL `https://arxiv.org/abs/2512.15943`.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 260–270, San Diego, California, 2016. Association for Computational Linguistics.

LangChain. Langgraph: Agent orchestration framework, 2024. URL https://langchain-ai.github.io.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Pierre Lison, Ildikó Pilán, David Sanchez, Montserrat Batet, and Lilja Øvrelid. Anonymisation models for text data: State of the art, challenges and future directions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4188–4203, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.323. URL https://aclanthology.org/2021.acl-long.323/.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

Nils Lukas, Ahmed Salem, Richard Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Beguelin. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (S&P)*, pp. 346–363. IEEE, 2023.

Ne Luo, Aryo Pradipta Gema, Xuanli He, Emile Van Krieken, Pietro Lesci, and Pasquale Minervini. Self-training large language models for tool-use without demonstrations. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1253–1271, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.69. URL https://aclanthology.org/2025.findings-naacl.69/.

Omri Mendels, Coby Peled, Nava Vaisman Levy, Sharon Hart, Tomer Rosenthal, Limor Lahiani, et al. Microsoft Presidio: Context aware, pluggable and customizable pii anonymization service for text and images, 2018. URL https://microsoft.github.io/presidio.

George R. Milne, George Pettinico, Fatima M. Hajjat, and Ereni Markos. Information sensitivity typology: Mapping the degree and type of risk consumers perceive in personal data sharing. *Journal of Consumer Affairs*, 51(1):133–161, Jun 2016. doi: 10.1111/joca.12111.

Kangyun Ning, Yisong Su, Xueqiang Lv, Yuanzhe Zhang, Jian Liu, Kang Liu, and Jinan Xu. Wtu-eval: a whether-or-not tool usage evaluation benchmark for large language models. *arXiv preprint arXiv:2407.12823*, 2024.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: large language model connected with massive apis. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Cheng Qian, Xinran Zhao, and Sherry Tongshuang Wu. "merge conflicts!" exploring the impacts of external distractors to parametric knowledge graphs. *arXiv preprint arXiv:2309.08594*, 2023.

Cheng Qian, Emre Can Acikgoz, Hongru Wang, Xiusi Chen, Avirup Sil, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. SMART: Self-aware agent for tool overuse mitigation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4604–4621, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.239. URL https://aclanthology.org/2025.findings-acl.239/.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xiangru Cong, Xiang Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark

Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world APIs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/28e50ee5b72e90b50e7196fde8ea260e-Paper-Conference.pdf.

Eidan J. Rosado. Pii-codex: a python library for pii detection, categorization, and severity assessment. *Journal of Open Source Software*, 8(86):5402, 2023. doi: 10.21105/joss.05402. URL https://doi.org/10.21105/joss.05402.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023.

Paul M Schwartz and Daniel J Solove. The pii problem: Privacy and a new concept of personally identifiable information. *New York University Law Review*, 86:1814–2011, 2011.

Freda Shi, Zheng Chen, Qiang Zhao, Chiyuan Zhang, et al. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31210–31227. PMLR, 2023.

Amy Steier, Andre Manoel, Alexa Haushalter, and Maarten Van Segbroeck. Nemotron-pii: Synthesized data for privacy-preserving ai, 2025. URL https://huggingface.co/datasets/nvidia/Nemotron-PII.

Maojun Sun, Ruijian Han, Binyan Jiang, Houduo Qi, Defeng Sun, Yancheng Yuan, and Jian Huang. A survey on large language model-based agents for statistics and data science. *The American Statistician*, pp. 1–14, 2025. doi: 10.1080/00031305.2025.2561140. URL https://doi.org/10.1080/00031305.2025.2561140.

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. Let me speak freely? a study on the impact of format restrictions on large language model performance. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1218–1236. Association for Computational Linguistics, 2024.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024. URL https://arxiv.org/abs/2403.08295.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, Guoyin Wang, and Chen Guo. GPT-NER: Named entity recognition via large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4257–4275, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.239. URL https://aclanthology.org/2025.findings-naacl.239/.

Weikai Xu, Chengrui Huang, Shen Gao, and Shuo Shang. Llm-based agents for tool learning: A survey. *Data Science and Engineering*, 10:533–563, 06 2025. doi: 10.1007/s41019-025-00296-9.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL `https://arxiv.org/abs/2407.10671`.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuan-Jing Huang. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8653–8665, 2023.

Fei Yu, Hongbo Zhang, Prayag Tiwari, and Benyou Wang. Natural language reasoning, a survey. *ACM Computing Surveys*, 56(12):1–39, 2024.

Pengfei Yu and Heng Ji. Information association for language model updating by mitigating lm-logical discrepancy. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pp. 117–129, Miami, FL, USA, 2024. Association for Computational Linguistics.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. Generate rather than retrieve: Large language models are strong context generators. In *International Conference on Learning Representations (ICLR)*, 2023.

A Appendix

B Benchmark Details

B.1 Full PII Type Distribution

Table 6 presents the complete ground-truth entity count for all PII types present in the main study benchmark (N = 2,000 samples). Counts are computed from the benchmark's `pii_codex_ground_truth` field in notebook `03_analyses_main.ipynb` (section 2.1, Full PII type distribution).

Table 6: Full ground-truth PII entity counts by type (main study, N = 2,000). Each count is the number of entity spans of that type in the benchmark; total matches Table 1.

PII Type	Entity count
PERSON	2,324
DATE_TIME	1,572
LOCATION	1,068
ADDRESS	862
EMAIL_ADDRESS	632
US_SOCIAL_SECURITY_NUMBER	632
PHONE_NUMBER	566
DATE	531
IP_ADDRESS	515
US_DRIVERS_LICENSE_NUMBER	383
GENDER	365
HEALTH_INSURANCE_ID	343
ZIPCODE	316
US_PASSPORT_NUMBER	379
CREDIT_CARD_NUMBER	137
LICENSE_PLATE_NUMBER	175
PASSWORD	127
OCCUPATION	102
URL	92
ABA_ROUTING_NUMBER	62
US_BANK_ACCOUNT_NUMBER	44
NRP	28
MEDICAL_LICENSE	30
RACE	18
AGE	18
SWIFT_CODE	17
MAC_ADDRESS	14
US_INDIVIDUAL_TAXPAYER_IDENTIFICATION	11
SEXUAL_PREFERENCE	7
Total (entity span count)	**11,370**

B.2 Labels Excluded from Scoring

The following 27 prediction labels appeared in model output but could not be unambiguously mapped to a PII-Codex canonical type and were excluded from scoring. They are listed here for transparency. `COUNTRY_OF_ORIGIN` was mapped to `LOCATION` and is not in this list.

```
ATTRIBUTE_NAME, ATTRIBUTE_VALUE, CLIENT_ID, COMMENTS, ID, IDENTIFIABLE,
IDENTIFICATION_NUMBER, IDENTIFIER, ID_NUMBER, INDIVIDUAL_ID,
```

`MEDICALCONDITION`, `NUMBER`, `PARENT_ID`, `PARTY`, `PERSONAL_IDENTIFIER`, `PERSON_ID`, `PII`, `PII_TYPE`, `PSYCHOLOGICAL_HISTORY`, `ROLE`, `SIBLING_ID`, `SUBSTANCEUSEDISORDERHISTORY`, `TEXT`, `TRAUMAHISTORY`, `US_ID`, `US_ID_NUMBER`, `VEHICLE_ID`

Labels such as `ATTRIBUTE_NAME`, `COMMENTS`, `ROLE`, and `TEXT` do not appear in any of the three source datasets and are model-output artifacts rather than true PII categories. `VEHICLE_ID` was considered for mapping to `LICENSE_PLATE_NUMBER` but was held for manual review and not included in the canonical mapping.

C Pilot Study Results

The pilot (n = 200) was run on the same hardware and under the same conditions as the main study, using a stratified subsample of the full benchmark. It served primarily to validate the pipeline and inform the main study configuration (parallelism, context bounds, timeouts). Pilot results are presented here as a citable record; formal conclusions are drawn from the main study only.

C.1 Pilot Summary Table

Table 7: Pilot study summary (n = 200, post-label alignment). f1_med = median F1; elapsed_μ = mean seconds per sample; turns_μ = mean conversation turns.

Model	Condition	f1_μ	prec_μ	rec_μ	f1_med	skill_viewed	elapsed_μ	turns_μ
gemma2_9b	with_docs	0.557	0.710	0.493	0.571	0	9.75	1.00
	with_skills	0.391	0.454	0.388	0.400	200	11.81	4.00
	with_tools	0.392	0.455	0.387	0.400	0	8.22	2.00
	zero_shot	0.530	0.683	0.466	0.571	0	7.60	1.00
llama3_8b	with_docs	0.412	0.446	0.419	0.453	0	8.56	1.00
	with_skills	0.384	0.419	0.409	0.400	200	15.74	3.93
	with_tools	0.433	0.467	0.456	0.437	0	14.91	2.00
	zero_shot	0.598	0.620	0.613	0.667	0	3.35	1.00
mistral_7b	with_docs	0.503	0.629	0.446	0.500	0	4.92	1.00
	with_skills	0.434	0.534	0.406	0.400	108	9.63	2.02
	with_tools	0.434	0.534	0.406	0.400	0	10.94	2.00
	zero_shot	0.497	0.628	0.438	0.571	0	3.04	1.00
qwen2_7b	with_docs	0.543	0.646	0.500	0.571	0	4.78	1.00
	with_skills	0.394	0.543	0.361	0.400	43	12.99	2.47
	with_tools	0.392	0.461	0.391	0.400	0	7.28	2.02
	zero_shot	0.556	0.634	0.528	0.608	0	3.01	1.00

C.2 Pilot Degradation Heatmap (Tabular)

The pilot directional pattern is consistent with the main study: tool and skills conditions underperform zero-shot for all models, and documentation is near-neutral.

Table 8: Pilot mean F1 delta vs. zero-shot by model and condition.

Model	Δ+Docs	Δ+Tool	Δ+Skills
gemma2_9b	**+0.028**	−0.138	−0.139
llama3_8b	−0.186	−0.165	−0.214
mistral_7b	**+0.005**	−0.063	−0.063
qwen2_7b	−0.013	−0.164	−0.162

D Full Statistical Tables

D.1 All Pairwise Comparisons

Table 9 extends Table 4 in the main text to include all six pairwise comparisons per model. Cohen's d and paired t and p are reported for all comparisons.

Table 9: Full pairwise delta summary (main study, N = 2,000 per model). % imp = percentage of samples where the condition outperforms the comparator.

Model	Comparison	t	p	d	% improved
llama3_8b	+Docs vs ZS	−17.94	< 0.0001	−0.401	24.3%
	+Tool vs ZS	−23.17	< 0.0001	−0.518	20.9%
	+Skills vs ZS	−29.94	< 0.0001	−0.670	18.1%
	+Tool vs +Docs	−1.83	0.067	−0.041	39.2%
	+Skills vs +Docs	−8.17	< 0.0001	−0.183	34.0%
	+Skills vs +Tool	−12.89	< 0.0001	−0.288	11.2%
mistral_7b	+Docs vs ZS	−1.98	0.048	−0.044	32.3%
	+Tool vs ZS	−17.59	< 0.0001	−0.393	26.7%
	+Skills vs ZS	−17.63	< 0.0001	−0.394	26.7%
	+Tool vs +Docs	−14.69	< 0.0001	−0.329	29.5%
	+Skills vs +Docs	−14.73	< 0.0001	−0.329	29.2%
	+Skills vs +Tool	−0.56	0.578	−0.012	1.6%
qwen2_7b	+Docs vs ZS	−0.61	0.543	−0.014	33.5%
	+Tool vs ZS	−24.45	< 0.0001	−0.547	19.9%
	+Skills vs ZS	−23.81	< 0.0001	−0.532	21.3%
	+Tool vs +Docs	−23.92	< 0.0001	−0.535	22.2%
	+Skills vs +Docs	−23.63	< 0.0001	−0.528	22.9%
	+Skills vs +Tool	+0.85	0.394	+0.019	26.6%
gemma2_9b	+Docs vs ZS	−0.63	0.530	−0.014	32.5%
	+Tool vs ZS	−20.99	< 0.0001	−0.469	23.3%
	+Skills vs ZS	−20.88	< 0.0001	−0.467	23.4%
	+Tool vs +Docs	−20.60	< 0.0001	−0.461	24.2%
	+Skills vs +Docs	−20.50	< 0.0001	−0.458	24.4%
	+Skills vs +Tool	+2.05	0.040	+0.046	1.3%

D.2 Confidence Interval Analysis

Table 10: Bootstrap 95% CI analysis (main study, target margin ±0.05). All cells sufficient at N = 2,000. req_n = minimum n for target margin.

Model	Condition	n	f1_μ	std	CI low	CI high	req_n
gemma2_9b	with_docs	2000	0.560	0.285	0.548	0.573	125
	with_skills	2000	0.400	0.249	0.389	0.411	96
	with_tools	2000	0.399	0.249	0.388	0.410	96
	zero_shot	2000	0.565	0.316	0.551	0.579	154
llama3_8b	with_docs	2000	0.454	0.338	0.439	0.469	177
	with_skills	2000	0.380	0.245	0.369	0.390	93
	with_tools	2000	0.437	0.257	0.426	0.449	102
	zero_shot	2000	0.615	0.319	0.601	0.629	157
mistral_7b	with_docs	2000	0.527	0.306	0.514	0.540	144
	with_skills	2000	0.406	0.260	0.394	0.417	105
	with_tools	2000	0.406	0.260	0.395	0.417	105
	zero_shot	2000	0.541	0.313	0.527	0.555	151
qwen2_7b	with_docs	2000	0.564	0.288	0.551	0.576	128
	with_skills	2000	0.375	0.262	0.364	0.387	106
	with_tools	2000	0.371	0.261	0.360	0.383	105
	zero_shot	2000	0.568	0.325	0.554	0.583	163

E Per-PII-Type Recall Tables

Table 11 presents recall by PII type and condition for all four models across the ten most frequent types (main study). Single-turn conditions (ZS, +Docs) and multi-turn (+Tool, +Skills) are combined; models are grouped within the table.

Table 11: Recall by PII type and model, all conditions (main study).

Model	PII Type	ZS	+Docs	+Tool	+Skills
Llama 3.1 8B	ADDRESS	0.68	0.04	0.56	0.50
	DATE	0.64	0.02	0.92	0.88
	DATE_TIME	0.53	0.68	0.02	0.02
	EMAIL_ADDRESS	0.93	0.82	0.98	0.95
	HEALTH_INSURANCE_ID	0.64	0.15	0.08	0.00
	IP_ADDRESS	0.85	0.63	0.96	0.94
	LOCATION	0.39	0.58	0.06	0.03
	PERSON	0.71	0.54	0.53	0.46
	PHONE_NUMBER	0.85	0.78	0.80	0.72
	US_SOCIAL_SECURITY_NUMBER	0.31	0.70	0.66	0.59
Mistral 7B	ADDRESS	0.50	0.05	0.49	0.49
	DATE	0.32	0.02	0.83	0.84
	DATE_TIME	0.29	0.51	0.00	0.00
	EMAIL_ADDRESS	0.93	0.91	0.97	0.97
	HEALTH_INSURANCE_ID	0.54	0.21	0.01	0.01
	IP_ADDRESS	0.76	0.80	0.95	0.95
	LOCATION	0.16	0.58	0.02	0.02
	PERSON	0.50	0.59	0.46	0.46
	PHONE_NUMBER	0.86	0.87	0.69	0.69
	US_SOCIAL_SECURITY_NUMBER	0.54	0.76	0.56	0.55
Qwen 2.5 7B	ADDRESS	0.61	0.03	0.51	0.46
	DATE	0.60	0.00	0.85	0.78
	DATE_TIME	0.33	0.78	0.00	0.00
	EMAIL_ADDRESS	0.97	0.98	0.96	0.97
	HEALTH_INSURANCE_ID	0.57	0.11	0.02	0.00
	IP_ADDRESS	0.81	0.90	0.93	0.86
	LOCATION	0.44	0.75	0.02	0.00
	PERSON	0.57	0.53	0.43	0.40
	PHONE_NUMBER	0.91	0.93	0.71	0.62
	US_SOCIAL_SECURITY_NUMBER	0.17	0.89	0.57	0.55
Gemma 2 9B	ADDRESS	0.53	0.00	0.51	0.51
	DATE	0.25	0.00	0.90	0.90
	DATE_TIME	0.26	0.62	0.00	0.00
	EMAIL_ADDRESS	0.97	0.98	0.99	0.99
	HEALTH_INSURANCE_ID	0.54	0.09	0.01	0.01
	IP_ADDRESS	0.71	0.86	0.98	0.98
	LOCATION	0.22	0.57	0.01	0.01
	PERSON	0.67	0.66	0.47	0.47
	PHONE_NUMBER	0.85	0.95	0.73	0.73
	US_SOCIAL_SECURITY_NUMBER	0.34	0.77	0.59	0.60

F Skill-Viewed Subgroup Analysis

F.1 Group sizes and mean F1

Table 12: Skill-viewed vs. not-viewed mean F1 (with_skills condition, main study). Llama and Gemma had 100% view rates; no not-viewed group is available for those models.

Model	Viewed n	Not-viewed n	View rate	Mean F1 difference
mistral_7b	948	1,052	47.4%	≈ 0
qwen2_7b	341	1,659	17.1%	≈ 0
llama3_8b	2,000	0	100%	n/a
gemma2_9b	2,000	0	100%	n/a

Analysis shows no discernible difference in mean F1 between viewed and not-viewed groups for Mistral or Qwen. This within-condition comparison controls for all experimental factors except document reading and provides the strongest available evidence that the Skill document itself does not drive performance.

F.2 Turn count and elapsed time correlations

Table 13: Pearson correlation of turns and elapsed seconds with F1 per sample (main study). n/a = single-turn condition; no turn variance.

	zero_shot		with_docs		with_tools		with_skills	
Model	turns	elapsed	turns	elapsed	turns	elapsed	turns	elapsed
gemma2_9b	n/a	−0.02	n/a	−0.01	n/a	**+0.05**	−0.01	**+0.06**
llama3_8b	n/a	−0.16	n/a	−0.04	−0.05	−0.03	−0.11	−0.08
mistral_7b	n/a	−0.16	n/a	−0.08	+0.00	**+0.08**	**+0.01**	**+0.07**
qwen2_7b	n/a	−0.07	n/a	**+0.02**	**+0.02**	**+0.11**	**+0.01**	**+0.09**

G Skill Document (`SKILL.md`)

The following is the `SKILL.md` document served to models via the `view_skill("pii-detection")` tool call in the with_skills condition. The same document was used for both pilot and main studies. It was compressed to approximately 150 tokens to prevent context overload in 7–9B models (see Section 3).

```
# pii-detection skill

## When to use
When asked to detect, identify, or redact personally identifiable
information (PII).

## Available tool
- 'analyze_pii': Detects PII entities and returns sanitized text.
  Call with: [TOOL_CALL: analyze_pii]

## PII types
The tool returns all PII types supported by PII-Codex (e.g. PERSON,
LOCATION, ADDRESS, DATE, DATE_TIME, PHONE_NUMBER, EMAIL_ADDRESS,
US_SOCIAL_SECURITY_NUMBER, US_PASSPORT_NUMBER, US_DRIVERS_LICENSE_NUMBER,
CREDIT_CARD_NUMBER, IP_ADDRESS, URL, ZIPCODE, and others). Use the tool
output as the source of truth for types and spans.

## Workflow
1. Call [TOOL_CALL: analyze_pii] to get detections and sanitized text
2. Format output as: JSON array of detections, then sanitized text

## Output format
[{"type": "PERSON", "text": "John Smith", "start": 0, "end": 10}]
Then the sanitized text with PII replaced by type labels like
[PERSON], [PHONE_NUMBER].
```

H Prompt Templates

The following prompt templates were used for each condition. All models received identical prompts; chat templates were applied per each model's tokenizer specification. `{text}` is replaced with the sample text at runtime; `{docs}` is replaced with PII-Codex reference documentation in the with_docs condition.

H.1 Zero-shot prompt

```
You are a PII (Personally Identifiable Information) detection expert.

## Task

Analyze the text below and identify all instances of PII. For each PII
span, determine its type, the exact text, and the character start and
end positions.

## Output Format

Return your findings as a JSON array only. Each item must have: type
(string), text (exact span), start (character index), end (character
index). If no PII is found, return an empty array.

## Text to Analyze

{text}
```

H.2 Documentation prompt (+Docs)

```
You are a PII (Personally Identifiable Information) detection expert.

## Background: PII-Codex Library

PII-Codex is a Python library for PII detection, categorization, and
severity assessment. It combines Microsoft Presidio detection with
academic frameworks for risk classification.

## When to Use This Skill

- Analyzing text or datasets for privacy risks before sharing
- Sanitizing user-generated content (social media posts, survey
  responses, etc.)
- Assessing compliance readiness for HIPAA, NIST, or DHS guidelines
- Preparing research data for publication
- Auditing logs or documents for accidental PII exposure

### PII Types Detected

- PERSON (names)
- EMAIL_ADDRESS
- PHONE_NUMBER
- LOCATION (addresses, cities)
- DATE_TIME (birthdates, specific dates)
- US_SSN (Social Security Numbers)
```

```
- CREDIT_CARD
- US_PASSPORT
- IP_ADDRESS
- URL
- PASSWORD (API keys, secrets, credentials)

### Risk Level Classification

PII-Codex categorizes detections on a 1-3 scale based on Schwartz &
Solove (2012):

| Level | Definition            | Examples                    |
|-------|-----------------------|-----------------------------|
| 1     | Non-Identifiable      | URLs, general locations     |
| 2     | Semi-Identifiable     | Partial addresses, age ranges|
| 3     | Identifiable          | Full names, SSNs, email     |

### Compliance Framework Mappings

Each detection is mapped to multiple compliance frameworks:

- NIST SP 800-122: Directly PII, Linked PII, Linkable PII
- HIPAA: Protected Health Information (PHI) identifiers
- DHS: Standalone PII vs Linkable PII
- Milne et al.: Information Sensitivity Typology clusters

## Best Practices

1. Always analyze before sharing - Run collection analysis on any
   dataset before external sharing
2. Check risk distribution - A high standard deviation indicates mixed
   sensitivity; review manually
3. Use sanitized_text for downstream tasks - The redacted output
   preserves structure while removing PII
4. Review Semi-Identifiable (Level 2) - These may become identifiable
   when combined with other data
5. Log detection frequencies - Patterns in PII types can reveal data
   collection issues

## Task

Analyze the provided text and identify all instances of Personal
Identifiable Information (PII).

For each PII instance found, provide:
1. The PII type (use the types listed above)
2. The exact text that contains the PII
3. The character positions (start and end) where the PII appears

## Output Format

Return your findings as a JSON array:
[{"type": "PII_TYPE", "text": "the actual PII text", "start": 0, "end": 10}]
```

```
If no PII is found, return an empty array: []

## Text to Analyze

{text}
```

H.3 Tool-augmented system message (+Tool)

The with_tools condition uses the LangGraph SkillsAgent with a single available tool (`analyze_pii`). The system message is:

```
You are a PII detection agent with access to the analyze_pii tool.

## Available Tool

**analyze_pii**: Detects Personal Identifiable Information in text
using PII-Codex.
- Returns: JSON with detected PII entities, types, positions, and risk
  levels

To call this tool, respond with EXACTLY:
[TOOL_CALL: analyze_pii]

After receiving tool results, use them to provide your final answer.

## Task

Analyze the following text for PII. Use the analyze_pii tool to get
accurate detections, then format the results.

## Text to Analyze

{text}

## Instructions

1. First, call the analyze_pii tool by responding with:
   [TOOL_CALL: analyze_pii]
2. After receiving results, format them as a JSON array with type,
   text, start, end for each PII found.
```

H.4 Skills-augmented system message (+Skills)

The with_skills condition uses the same LangGraph agent with three tools: `list_skills`, `view_skill`, and `analyze_pii`. The system message is:

```
You are a PII detection agent. Detect all PII in the text below.

You have three tools. Call them in order using exact bracket syntax:
1. [TOOL_CALL: list_skills]
2. [TOOL_CALL: view_skill pii-detection]
3. [TOOL_CALL: analyze_pii]
```

```
Your first reply must be exactly: [TOOL_CALL: list_skills]

After receiving analyze_pii results, output:
1. JSON array: [{"type": "PII_TYPE", "text": "...", "start": 0, "end": 10}]
2. Sanitized text with placeholders like [PERSON], [PHONE_NUMBER]

If no PII is found, return [].

## Text to Analyze

{text}
```