AGENTIC PRIVATE EVOLUTION: DIFFERENTIALLY PRIVATE SYNTHESIS OF AGENT WORKSPACE DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluating enterprise agents requires realistic workspace data, but private email threads, conversations, and documents cannot be freely shared. Private Evolution (PE) offers differentially private (DP) synthesis without training the generator. Applying PE to workspace data, however, presents three challenges: independent generation lacks shared context whose rules vary across data types; semantic similarity does not ensure structural validity; and vote-based selection can lose coverage of requested sample types. To address these challenges, we propose **Agentic Private Evolution (Agentic-PE)**, a framework for DP synthesis of agent environment data. Starting from a DP attribute table that specifies the samples to generate, Agentic-PE introduces three complementary modules: a constraint module provides shared participants and data-type-specific context rules; a validation agent checks and repairs candidates before private voting; and constrained selection retains one feasible candidate per table row to preserve coverage. The modules operate on DP outputs and synthetic samples, adding no privacy cost. On email at a matched privacy budget, Agentic-PE achieves 6.4 times Aug-PE's semantic recall at lower Fréchet Inception Distance (FID) and raises structural validity from 62.8% to 93.1%. Its corpus improves downstream training over a DP fine-tuning baseline and yields positively correlated retrieval and reply-model rankings on real and synthetic benchmarks. The framework also improves recall and FID over matched-budget Aug-PE on chat, meetings, and internal documents.

1 INTRODUCTION

Enterprise agents process email, search conversations, and work with internal documents. An email assistant summarizes threads, retrieves information, and drafts replies (Microsoft, 2026; Google, n.d.). Evaluating these capabilities requires realistic workspace records with consistent participants and context. Sensitive user and business information limits the sharing and reuse of real workspaces. Differentially private (DP) synthesis offers substitute data with a formal privacy guarantee (Dwork & Roth, 2014).

DP stochastic gradient descent (DP-SGD) can train a private text generator (Abadi et al., 2016; Yue et al., 2023), but requires access to model weights. Private Evolution (PE) instead generates candidates through an API, selects them using noisy private similarity votes, and varies the survivors (Lin et al., 2024). Aug-PE extends this approach to text (Xie et al., 2024).

However, applying PE to agent environment data presents three challenges. ① **Independent generation lacks shared context.** Separate procurement threads may involve the same buyer and manager, but independent generation provides no shared participant roster or reporting relationships. Context rules also vary by data type: email needs recipients, chat needs channel membership, and meetings need a chair and attendees. ② **Similarity does not ensure validity.** A procurement email can resemble private correspondence while naming an unknown sender or dating a reply before its request. High similarity votes do not establish structural validity. ③ **High votes do not ensure coverage.** Selecting the highest-voted candidates can retain several variants of one sample type while dropping another. DP metadata improves PE initialization (Chien et al., 2026), but does not by itself ensure that later rounds preserve the requested mix.

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

We introduce **Agentic Private Evolution (Agentic-PE)**, which addresses these challenges with three modules (Figure 1). First, a *DP attribute table* specifies the requested samples, such as a medium-length procurement thread expecting a reply. A *constraint module* builds shared fictional context with constraints tailored to each data type: sender–recipient relationships for email, channel membership for chat, chairs and attendees for meetings, and authors and approvers for documents. A *validation agent* checks and repairs candidates before private voting. Finally, *constrained selection* retains one feasible candidate per indexed table row, preventing multiple variants of one row from displacing another. Survivors are varied in the next round while the table and shared context remain fixed. These modules read only DP outputs and synthetic samples, adding no privacy cost. We also compare candidates through embeddings of canonical topic summaries, so topic similarity is assessed separately from structural checks.

Overall, our contributions are threefold. (i) We propose **Agentic Private Evolution (Agentic-PE)** for DP synthesis with shared context, validation, and coverage constraints (§3). (ii) We conduct extensive experiments demonstrating improved semantic fidelity, structural validity, and downstream utility on email (§4.2–§4.3). (iii) We extend Agentic-PE to chat, meetings, and internal documents, demonstrating improved recall and FID over matched-budget Aug-PE across all three data types (§4.4).

2 Related work

Differentially private text generation. DP-SGD fine-tunes language models on private data before sampling synthetic text (Abadi et al., 2016; Yue et al., 2023; Mattern et al., 2022; Kurakin et al., 2023). Parameter-efficient variants reduce training cost (Yu et al., 2022; Hu et al., 2022); strong DP fine-tuning results and practical guidance are given by Li et al. (2022) and Ponomareva et al. (2023). These approaches require model weights. In-context alternatives privatize predictions (Wu et al., 2024), generate DP few-shot demonstrations (Tang et al., 2024), or construct instruction sets (Yu et al., 2024). Our target is a corpus of structured workspace samples that can support both training and evaluation. We also draw on DP tabular synthesis, which releases low-order attribute distributions and generates a table consistent with them (Zhang et al., 2017; McKenna et al., 2021; 2022). In Agentic-PE, that table provides generation conditions for text. Privacy accounting uses standard Gaussian composition tools (Mironov, 2017; Balle & Wang, 2018; Gopi et al., 2021).

Private Evolution. PE replaces private model training with noisy nearest-neighbor voting over API-generated candidates (Lin et al., 2024). Aug-PE adapts the procedure to text (Xie et al., 2024); PrE-Text targets private federated text (Hou et al., 2024), and Sim-PE uses simulators for image synthesis (Lin et al., 2025). MAPLE is particularly relevant: it uses AIM (McKenna et al., 2022) to synthesize DP metadata, then combines that metadata with in-context examples to improve PE initialization (Chien et al., 2026). Agentic-PE carries the attribute table through later rounds and adds a shared constraint module, candidate validation, and coverage constraints on selection. Our evaluation therefore measures both the semantic fidelity of individual samples and the structural properties of the resulting collection, complementing work on structured synthetic-text evaluation (Wang et al., 2025; Sun et al., 2025).

Agentic synthetic data generation. Language-model pipelines can plan, generate, and filter synthetic training data (Long et al., 2024; Wang et al., 2023; Xu et al., 2025; Mitra et al., 2024), use personas to broaden coverage (Ge et al., 2024), or simulate recurring participants across episodes (Park et al., 2023). Workspace benchmarks similarly require environments with related records and actionable structure (Drouin et al., 2024; Styles et al., 2024; Yao et al., 2024). Agentic-PE combines shared context, validation, and coverage-preserving selection in private corpus synthesis: every conditioning signal derived from the private corpus passes through a DP mechanism, after which the agents construct and reuse a fictional workspace.

3 Agentic Private Evolution

Figure 1 and Algorithm 1 show the loop. We define the generation objects (§3.1) and three agentic modules (§3.2), then introduce the two privacy channels through which private information enters the synthesis process: attribute release and private similarity voting (§3.3).

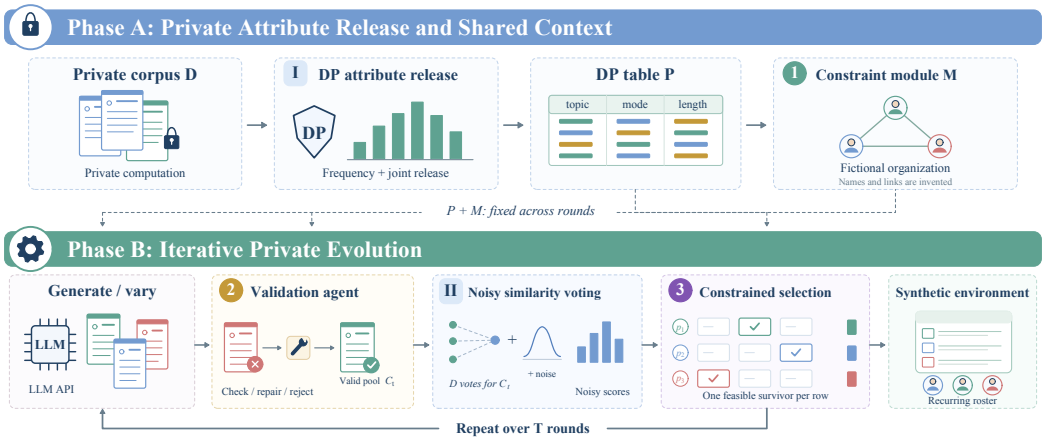


Figure 1: Agentic Private Evolution framework. Phase A: attribute release and shared-context construction. **Phase B**: generation/variation, validation, private voting, and selection over T rounds. Circled Arabic numerals denote our **three agentic modules**: **1**, the constraint module M; **2**, validation and repair; and **3**, constrained selection of one feasible survivor per table row. Roman numerals denote the **two privacy channels**: **I**, DP attribute release producing P, and **II**, noisy similarity voting on C_t. Dashed arrows carry the fixed conditions P and M; the bottom feedback arrow returns selected survivors to the next round.

3.1 Problem Statement and Private Evolution

Let $D = \{x_1, \ldots, x_n\}$ be a private corpus of one data type, where a sample is an email thread, chat conversation, meeting agenda item, or document. Two corpora are neighbors if one adds or removes a single sample, which may contain multiple messages. We seek a synthetic corpus $\tilde{D}$ of size m that satisfies (ε, δ)-DP with respect to D, whose samples pass the stated validity checks and whose collection follows the released attribute targets. By *population structure* we mean properties of that collection, such as participant recurrence and the distribution of sample attributes, including communication mode and length. Participant identities and relationships are fictional; the released attributes guide the statistical targets for the synthetic collection.

An *attribute* describes a property of a sample, such as its topic, communication mode, or length category. Let $\mathcal{A}$ be a schema of k categorical attributes with fixed allowed values, and $a(x)$ their values for sample x. The *DP attribute table* $P = (p_1, \ldots, p_m)$ specifies the m samples to generate: each column is an attribute, and each row p_i is one output sample's target combination. For example, a row may request a medium-length, interactive email thread about procurement. Rows with identical attributes still have distinct indices and correspond to separate output samples. $\tilde{H}$ denotes the marginal distributions read from P, and M the constraint module containing shared participants and rules. The generation API provides $\mathrm{Gen}(p, c)$ to draft a sample with attributes p in context c, and $\mathrm{Var}(x, p, c)$ to draft a variation of x.

Let $\phi(x)$ be the embedding used to compare samples. Unlike Aug-PE, which encodes the original text directly (Xie et al., 2024), we encode a short topic summary of x. This emphasizes subject matter; validation separately checks format and content consistency. For private samples, this preprocessing stays inside the data holder's protected environment, called the trusted boundary.

Standard PE uses an API to initialize candidates, release noisy nearest-neighbor votes, select survivors, and generate variations (Lin et al., 2024; Xie et al., 2024). Private information enters through T noisy vote histograms; generation and selection are post-processing. MAPLE adds DP metadata for initialization (Chien et al., 2026). Agentic-PE reuses P and M throughout evolution, validates candidates before voting, and retains one per table row. Only the attribute release adds privacy cost; the agents do not access private samples.

3.2 Three Agentic Modules

(1) The constraint module: shared context for each data type. An agent reads $\tilde{H}$ and P to construct a shared module M containing participant profiles, roles, relationships, and participation

rules. For email, M defines a fictional organization with names, job titles, departments, and reporting relationships, together with rules for choosing senders and recipients. Threads and their later variations reuse this organization, so independently generated samples share consistent participants and context.

The constraints differ by data type: email uses recipient and broadcast rules; chat uses channels with defined purposes and membership; meetings use committees with chairs, members, and meeting schedules; documents use authors and approvers. Data-type adapters implement these rules while reusing the evolution loop (Table 8).

(2) The validation agent: checking candidates before voting. Conditioning on M does not ensure that a generated sample satisfies it. A thread may refer to a person outside the roster, answer a nonexistent message or place a reply before the request. Similarity voting does not check these rules: a malformed but semantically close candidate can still receive votes and survive.

Before voting, the validation agent checks format, participant consistency with M, and semantic consistency within and across samples. Format checks cover required fields and message structure; content checks cover reply order, participant roles, calculations, dates, and agreement with previously established facts. The agent identifies a failed check and requests a local correction, such as fixing an inconsistent title or total, while preserving participants and message structure. The corrected sample is checked again; unresolved content failures are discarded (Appendix G).

(3) Constrained selection: one feasible survivor per table row. Traditional greedy top-m selection retains the m candidates with the highest noisy vote counts, but can keep several variants of one attribute row while dropping another. For example, two output rows may request a procurement exchange and a policy announcement; the two highest-voted candidates may both be procurement variants, leaving no policy announcement. Our *selection agent* instead reads the candidates, noisy votes, P, and M and selects one feasible survivor per row, prioritizing votes subject to attribute, participant, and interaction constraints.

Each candidate carries the index of the table row used to generate its ancestor. Candidates with row index i form $C_t^{(i)}$, so $C_t = \bigcup_{i \le m} C_t^{(i)}$. Let $A_i \subseteq C_t^{(i)}$ contain the candidates allowed by the constraints given earlier choices. When each A_i is nonempty, the selected set is

$$S_t \;=\; \{c_1^\star, \ldots, c_m^\star\}, \qquad c_i^\star \;=\; \arg\max_{c \,\in\, A_i} \; \tilde{v}_t[c], \tag{1}$$

$$A_i \subseteq C_t^{(i)} : \quad \underbrace{\text{row's mode matched}}_{\text{attribute match}}, \qquad \underbrace{\max_u \text{share}_u \le \tau_u}_{\text{participant cap}}, \qquad \underbrace{|\rho(S) - \rho(P)| \le \tau_\rho}_{\text{interactive mix}},$$

where share_u is the fraction of samples authored by participant u and $\rho(\cdot)$ the fraction admitting a reply. The bound τ_u limits any one participant's share, and τ_ρ is the allowed difference in reply fraction between the selected set S and the attribute table P. A_i also drops candidates carrying a detected defect or repeating a duplicate group, template or topic already kept. The private vote orders candidates, while P and M define which are allowed. The selection agent proposes survivors, narrowing A_i to its admissible proposals. A deterministic check verifies the proposed set; if the agent fails or violates the constraints, the same selection rule is applied by a deterministic fallback (Appendix F). The selection ablation evaluates this shared rule. Under the stated feasibility and variation assumptions, Appendix D establishes row coverage (Proposition 2) and analyzes convergence (Theorem 2).

3.3 Two Privacy Channels

Agentic-PE accesses private samples through two privacy channels: the attribute release and similarity voting. Each channel can involve several DP mechanisms. The attribute channel first computes noisy frequency counts when needed, such as the number of samples per sender. This *frequency pre-release* uses T_f steps, followed by T_r tabular synthesis steps. The voting channel releases one noisy histogram in each of T text-generation rounds. All of these steps use Gaussian mechanisms and are accounted for together in Proposition 1.

1) Attribute release. Private samples are represented by rows in the fixed categorical schema. For a fixed output of the frequency pre-release, the conceptual joint count $H(\mathcal{A}) = \sum_{x \in D} \mathbf{e}_{a(x)}$ counts

Algorithm 1 Agentic Private Evolution. Stages marked [AGENT] read DP outputs and synthetic samples only, never D. Here p_x is the attribute row assigned to x, and Var_j produces its jth variation, with b variations per survivor.

Require: private corpus D, budget (ε, δ), size m, rounds T

$\sigma \leftarrow$ noise multiplier solving the composition in Proposition 1

$\tilde{f} \leftarrow \textsc{DPFrequencies}(D, \sigma)$ ▷ T_f steps; $\mathcal{A}$'s corpus-level attribute is a threshold of $\tilde{f}$

$\tilde{H}, P \leftarrow \textsc{DPRelease}(D, \mathcal{A}, \sigma)$ ▷ privacy channel 1; T_r steps, m indexed table rows

$M \leftarrow \textsc{BuildModule}(\tilde{H}, P)$ ▷ [AGENT] constraint module

$S_0 \leftarrow \textsc{Validate}\big(\{\mathrm{Gen}(p, M) : p \in P\},\ M\big)$ ▷ [AGENT] participants, [AGENT] validation

for $t = 1$ to T **do**

  $C_t \leftarrow \textsc{Validate}\big(\{\mathrm{Var}_j(x, p_x, M) : x \in S_{t-1},\ j \le b\},\ M\big)$ ▷ [AGENT] validation; b variations each

  $\tilde{v}_t \leftarrow \sum_{x \in D} \mathbf{e}_{\arg\min_{c \in C_t} \|\phi(x) - \phi(c)\|} + \mathcal{N}(0, \sigma^2 I)$ ▷ privacy channel 2

  $S_t \leftarrow \textsc{Select}(C_t, \tilde{v}_t, P, M)$ ▷ [AGENT] selection, Eq. 1; verified fallback

end for

return $\tilde{D} \leftarrow S_T$

samples in each *attribute cell*, one combination of attribute values. The vector $\mathbf{e}_{a(x)}$ has a one at sample x's cell and zeros elsewhere. Adding or removing one sample changes this count vector by at most one in Euclidean norm; this maximum change is its ℓ_2 sensitivity. Tabular PE approximates this attribute distribution through T_r noisy nearest-neighbor votes over candidate rows (Lin et al., 2024) and releases the synthetic table P. Its marginals $\tilde{H}$ and attribute combinations provide explicit generation targets. Modeling the attributes jointly retains information about their co-occurrence. The released table contains categorical attributes, while M supplies fictional participants and relationships.

2) Similarity voting. In round t with candidate pool C_t, each private sample votes for its nearest candidate, $v_t = \sum_{x \in D} \mathbf{e}_{\arg\min_{c \in C_t} \|\phi(x) - \phi(c)\|}$, and we publish $\tilde{v}_t = v_t + \mathcal{N}(0, \sigma^2 I)$, where σ is the noise standard deviation. One sample moves one unit vote, so $\Delta_2 v_t = 1$. This histogram is an aggregate statistic: it measures how many private samples are closest to each proposed candidate. It can indirectly favor common topics or styles, but its bins index candidates, not explicit attribute targets. The attribute table, shared roster, and constrained selection impose those requirements alongside the vote-based preference.

Proposition 1 (Composition of the Gaussian steps)**.** *Fix the schema $\mathcal{A}$ and the index set of the frequency vector, both independent of D. Agentic-PE uses $N = T_f + T_r + T$ Gaussian steps: T_f for the frequency pre-release, T_r for the attribute release, and T for private voting. Conditional on earlier releases, each query has ℓ_2 sensitivity at most 1 under add/remove sample adjacency and adds fresh noise $\mathcal{N}(0, \sigma^2 I)$. Let (ε, δ) be the privacy bound for the composition of these N Gaussian steps at noise scale σ (Appendix D.1). The released synthetic corpus is then (ε, δ)-DP, provided the agents read only DP outputs and synthetic samples.*

Generation, repair, and selection are post-processing of these releases and incur no additional privacy cost. We fix (ε, δ) and solve for a shared σ. Email uses $T_f = 1$ because its sender-frequency category depends on other samples: adding a thread can change the frequency category of earlier threads from the same sender. Releasing noisy sender counts first makes the subsequent attribute query sensitivity one conditional on that release (Lemma 1). Data types with only per-sample attributes use $T_f = 0$. Appendix C states the fixed-index assumption, and Appendix D gives the privacy proof and utility analysis.

3.4 Implementation

Each data type uses nine fixed categorical attributes, obtained by annotation and structural measurements inside the trusted boundary. Tabular PE releases m indexed rows over $T_r = 15$ steps. Email adds one DP sender-frequency step. Each survivor produces $b = 2$ variations; parents are not retained. Validation allows two local repair attempts. For a batch of B drafts, the round stops if more than $\max(3, 0.02B)$ still fail content checks after repair; otherwise, failed candidates are removed. Appendices B, F, and G give the adapters, thresholds, and checks.

Table 1: Component ablation on email. Ten-round means, 400 synthetic against 2,000 private threads. *attrs.* is the number of released attributes, *module* whether the constraint module is used, *vote* whether selection reads the private vote. *valid* is the deterministic validity rate and *shape JS* the mean divergence of derived attribute distributions between synthetic and private samples. The last row is what two disjoint halves of the private corpus score under the same estimator and sample sizes.

Arm	attrs.	module	vote	P ↑	R ↑	FID ↓	valid ↑	shape JS ↓
Aug-PE	none	×	✓	0.586	0.064	0.1009	0.628	0.261
+ base schema	7	✓	✓	0.544	0.286	0.0542	**0.955**	0.357
+ derived attrs., no module	9	×	✓	0.546	0.391	0.0472	0.786	0.330
Agentic-PE, random selection	9	✓	×	0.582	0.335	0.0471	0.895	0.098
Agentic-PE (full)	9	✓	✓	**0.592**	**0.412**	**0.0455**	0.931	**0.097**
private reference	–	–	–	*0.897*	*0.896*	*0.0063*	*1.000*	*0.000*

All arms use $\varepsilon = 5.2985$ and $\delta = 10^{-5}$, with the noise multiplier calibrated to each arm's mechanisms. The DP-SGD synthesis baseline is compared separately in Table 2 and §4.3.

Table 2: Semantic fidelity and structural validity on email. Each corpus contains 400 synthetic samples and is scored against the same 2,000 private samples with the estimator and checks used in Table 1. Agentic-PE uses $\varepsilon = 5.2985$ and DP-SGD uses $\varepsilon \leq 5.293$, both at $\delta = 10^{-5}$. P and R denote semantic precision and recall; *valid* is the structural validity rate. Bold marks the best value in each column.

Corpus	P ↑	R ↑	FID ↓	valid ↑
Agentic-PE (ten-round mean)	0.592	**0.412**	**0.0455**	**0.931**
DP-SGD 7B	**0.882**	0.183	0.0839	0.555
Untrained 7B	**0.882**	0.140	0.0900	0.682

Agentic-PE reports the ten-round mean from Table 1. DP-SGD generates one corpus after private fine-tuning; the untrained model generates one corpus without fine-tuning.

4 Experiments

4.1 Setup

We use email as our primary testbed, synthesizing data from 2,000 Avocado threads (Oard et al., 2015) and holding out 174 separate real threads for downstream evaluation. We also use 2,000 Slack conversations, 232 QMSum agenda items (one per meeting), and 284 Enron documents (Chatterjee et al., 2020; Zhong et al., 2021; EDRM, 2013). Unless stated otherwise, we configure email experiments with $(\varepsilon, \delta) = (5.2985, 10^{-5})$, $m = 400$, and $T = 10$; we recalibrate each ablation's noise to match the full privacy budget. The DP-SGD baseline uses $\varepsilon \leq 5.293$ at the same δ.

We compute semantic precision, recall, and FID from canonical topic summaries embedded with Sentence-T5 (Ni et al., 2022; Kynkäänniemi et al., 2019; Heusel et al., 2017). Validity measures deterministic structural checks; shape JS is the mean Jensen–Shannon divergence of sender-frequency and message-length categories. We match estimator sample sizes and construct same-distribution references from disjoint private subsets.

We evaluate two downstream benchmark tasks: **auto-reply**, generating the mailbox owner's next reply from an email thread prefix, and **retrieval**, finding email or document evidence relevant to a query. We evaluate synthetic-data training through held-out real-email prediction (RQ3), and benchmark utility through real–synthetic ranking agreement for reply models and retrieval systems (RQ4). Appendix B.1 gives the full protocol and metric definitions, and Appendix C gives preprocessing.

4.2 Semantic fidelity and structural validity

RQ1: comparison with DP synthesis baselines. To assess whether Agentic-PE improves synthesis quality under a fixed privacy budget, we compare it with API-based Aug-PE and a DP-SGD generator. We observe substantially higher recall and lower FID: Agentic-PE achieves 0.412 and 0.0455, compared with 0.064 and 0.1009 for Aug-PE, and 0.183 and 0.0839 for DP-SGD (Ta-

bles 1–2). Precision remains similar to Aug-PE (0.592 versus 0.586), suggesting that the recall gain reflects broader coverage without a corresponding loss of semantic precision.

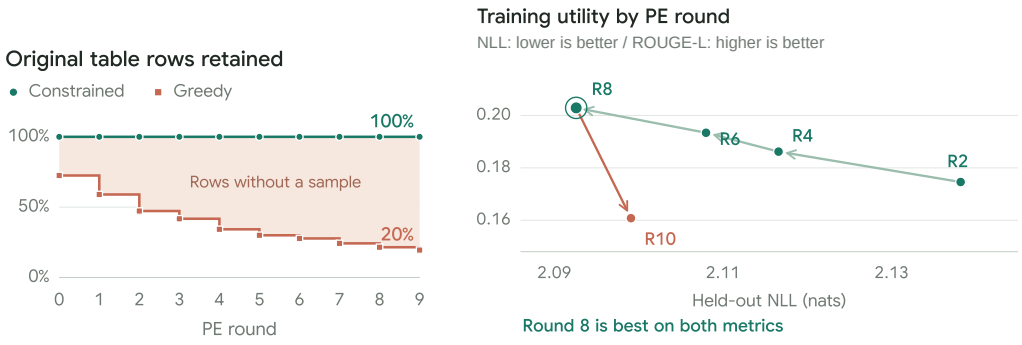
We further examine whether semantic similarity coincides with structural validity. Agentic-PE reaches validity 0.931, compared with 0.628 for Aug-PE and 0.555 for DP-SGD. Despite its lower validity, DP-SGD achieves precision 0.882, matching the untrained model to three decimals. This discrepancy may arise because topic-summary embeddings place plausible professional prose near private topics even when participant or thread structure is invalid. We therefore evaluate coverage and validity alongside precision.

RQ2: how do the components change corpus quality? To identify which design choices support these gains, we ablate attribute specification, shared context with validation, and selection under matched privacy budgets (Table 1).

Effect of derived attributes. We augment the base schema with sender frequency and mean message length to test whether explicit structural targets improve coverage. We observe that recall increases from 0.286 to 0.412, while final-round sender-frequency divergence falls from 0.837 to 0.072 and the sender count increases from 13 to 81. These results show that explicit structural targets improve participant coverage and sender-frequency alignment.

Effect of shared context and validation. Holding the nine attributes fixed, we add the constraint module and its checks to evaluate their joint contribution. We find that validity increases from 0.786 to 0.931, and shape JS decreases by 3.4×. These improvements are consistent with shared context supplying cross-sample rules that attribute conditioning alone does not enforce.

Effect of voting and constrained selection. To isolate the role of private votes, we compare against random selection with the same release, module, variation mode, seed, candidate budget, and privacy budget. We find that voting improves ten-round mean precision by 0.075 and lowers mean FID by 0.0013 (Figure 6), indicating useful similarity guidance. We then test whether greedily following the same votes preserves coverage. By the final round, greedy selection retains only about 20% of the original rows and 41 senders, whereas constrained selection retains all rows and 81 senders (Figure 2a). Its recall is 0.098 higher, with precision 0.065 lower. Together, these results suggest complementary roles: votes favor semantically close candidates, while row constraints prevent this preference from concentrating the population on a subset of generation targets.



(a) **(b)**

Figure 2: Coverage and downstream utility across rounds. (a) With the same votes, greedy selection retains about 20% of original table-row indices by round ten; constrained selection retains all. Coverage is the fraction of original row indices retained. **(b)** Models trained on rounds 2, 4, 6, 8, and 10 are evaluated on the same 174 held-out real threads. Round 8 gives the best NLL and ROUGE-L.

Sensitivity and repetition. To examine budget sensitivity, we evaluate six privacy budgets, using one seed each. Within this sweep, we observe that increasing ε from 1 to 3 raises recall from 0.223 to 0.396 and lowers FID from 0.0618 to 0.0495. At $\varepsilon = 10$, we obtain FID 0.0473, compared with 0.0477 for the no-noise reference. We observe little utility change beyond $\varepsilon \approx 3$ (Figure 5a, Table 5), suggesting diminishing utility returns from further relaxing privacy in this setting. We further repeat synthesis across five independent runs and observe decreasing FID during the first two rounds, including under three representations not used for voting (Figure 5b); the early gains thus extend beyond the voting representation.

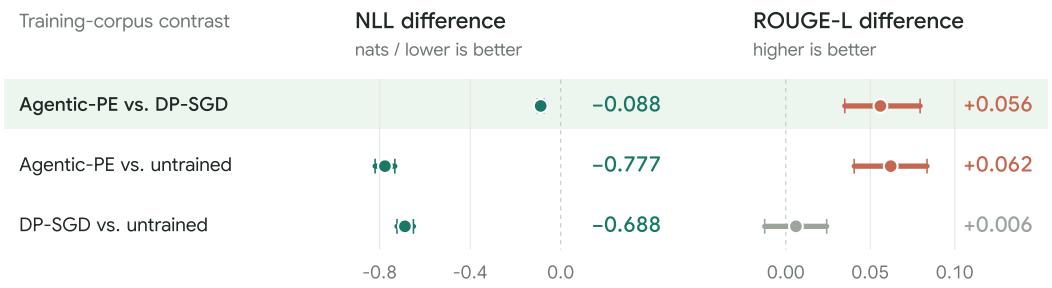


Figure 3: RQ3: training on synthetic email. Paired 95% bootstrap intervals on 174 held-out real threads (4,000 resamples). Agentic-PE improves NLL and ROUGE-L over DP-SGD synthesis; both intervals exclude zero. DP-SGD does not separate from no training on ROUGE-L (gray).

4.3 Downstream Training and Evaluation

RQ3: does synthetic training improve performance on real email? To test whether improved synthesis quality benefits real-data prediction, we fine-tune Qwen2.5-7B-Instruct with LoRA (Qwen Team, 2024; Hu et al., 2022) on each synthetic corpus using the same recipe and evaluate on 174 held-out real threads. We extract 304 usable prompt–target pairs from 400 Agentic-PE threads, compared with 58 from DP-SGD, consistent with Agentic-PE's higher validity.

Table 3: Downstream utility on held-out real email. Qwen2.5-7B-Instruct fine-tuned on each synthetic corpus, evaluated on 174 held-out *real* threads; identical recipe, the training corpus is the only difference. *ex.* counts the next-message training examples constructible from 400 generated threads.

Training corpus	NLL ↓	ROUGE-L ↑	ex. ↑
Agentic-PE corpus (round 10)	**1.8448**	**0.2564**	**304**
Agentic-PE corpus (round 2)	1.8662	0.2436	304
DP-SGD corpus (7B)	1.9236	0.2004	58
Untrained	2.5703	0.1943	–

We observe lower NLL and higher ROUGE-L for Agentic-PE than for DP-SGD, with both paired 95% intervals excluding zero (Figure 3, Table 3). Together, these results demonstrate that Agentic-PE produces more usable training examples and improves prediction on held-out real email.

We further compare checkpoints across synthesis rounds and identify round 8 as the best-performing checkpoint on both NLL and ROUGE-L (Figure 2b). We report additional auto-reply fine-tuning results in Appendix A.

RQ4: comparing systems on real and synthetic benchmarks.

To assess whether synthetic data supports system selection, we compare retrieval and reply-model rankings on real and synthetic benchmarks.

Retrieval. We compare eleven email retrieval systems spanning BM25 (Robertson & Zaragoza, 2009), dense retrieval, hybrid fusion, LLM rerankers, and agentic RAG (Lewis et al., 2020), and nine document retrieval systems. We observe positive rank correlations between real and synthetic benchmarks: $\rho = 0.659$ for single-thread email evidence, 0.618 for two-thread evidence, and 0.898 for documents (Figure 4, Table 6). We generate queries and relevance labels from each corpus using the same procedure and seed (Appendix H). The positive ranking agreement across all three benchmarks supports using synthetic corpora for comparative retrieval evaluation.

Reply-model selection. To test whether ranking agreement extends to generation, we select ten candidate LLMs and prompt each to generate the mailbox owner's next reply from 10 real and 10 synthetic email thread prefixes. We score the replies with a pre-registered seven-dimension judge (Zheng et al., 2023) and obtain $\rho = 0.733$, identifying the same winning model on both benchmarks (Figure 9). We also observe positive rank correlations with all three independent judges: 0.733 for GPT-5.4, 0.697 for GPT-5.6 Sol, and 0.689 for Mistral Large 3 (Appendix I). By rescoring the same stored replies, we test whether ranking agreement persists across evaluator choices. The

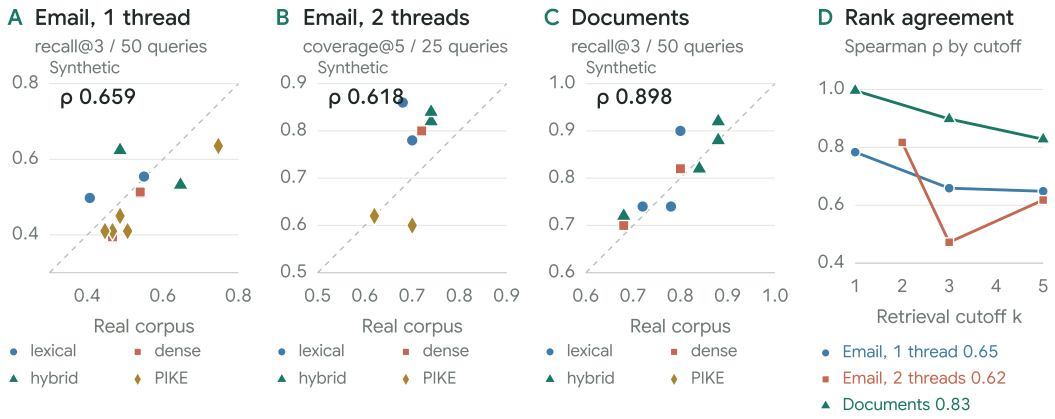


Figure 4: RQ4, retrieval over the three benchmarks (50/25/50 queries, 11/11/9 systems). Retrieval systems scored on the real corpus (x) and the Agentic-PE corpus (y), each on its own corpus-grounded benchmark built by one generator (Appendix H); the dashed line is exact agreement. **Right**: rank agreement against the cutoff at which the leaderboards are compared.

Table 4: Transfer beyond email. *Left*: scores divided by each corpus's same-distribution reference; 1.0 matches that reference score. *Right*: differences from Aug-PE at matched $(\varepsilon, \delta) = (4.4318, 10^{-5})$, 60 samples, three rounds, and the same voting representation. Means over four snapshots.

Corpus	measured / floor			valid	Agentic-PE − Aug-PE, matched ε		
	FID ↓	prec. ↑	rec. ↑	rate ↑	ΔFID ↓	Δprec. ↑	Δrec. ↑
Chat	2.9×	0.63	0.22	0.813	−0.054	−0.092	+0.108
Meetings	2.2×	0.62	0.37	0.988	−0.024	−0.058	+0.181
Documents	**1.8×**	**1.00**	0.36	0.966	−0.015	−0.001	+0.119

Each Aug-PE noise multiplier is recalibrated for its three voting rounds. Agentic-PE improves recall and FID in all three corpora, with lower precision.

consistently positive correlations support using synthetic workspace data for comparative reply-model evaluation.

4.4 Transfer across data types

RQ5: semantic and structural behavior beyond email. To evaluate applicability beyond email, we adapt the shared context and validation rules to chat, meeting agenda items, and internal documents while retaining the evolution loop (Table 8). At matched privacy budgets, we observe recall gains over Aug-PE by 0.108, 0.181, and 0.119, respectively, together with lower FID in all three cases (Table 4). The corresponding precision changes are −0.092, −0.058, and −0.001.

We additionally normalize scores by each corpus's same-distribution reference to compare fidelity relative to a common reference scale. We find that documents reach the reference precision and 1.8× its FID, whereas chat and meetings reach 2.9× and 2.2× the reference FID. Across all three additional data types, we obtain higher recall and lower FID than matched-budget Aug-PE while retaining the same evolution framework and adapting the context and validation rules.

5 Conclusion

We introduced Agentic-PE for differentially private workspace synthesis with shared context, validation, and coverage-preserving selection. At a matched privacy budget, we achieve 6.4× Aug-PE's email recall and increase structural validity from 62.8% to 93.1%. Across four data types, our results demonstrate improved synthesis quality and support using synthetic workspace data for downstream training and comparative agent evaluation.

AI USE STATEMENT

Language models are used for the data preparation, annotation, synthetic generation, validation/repair, selection, and evaluation procedures described in this paper. Generative AI also assisted with manuscript editing, organization, citation checking, clarification of theoretical assumptions, and figure and LaTeX preparation. The authors are responsible for verifying all AI-assisted text, claims, and artifacts.

ETHICS STATEMENT

The corpora are public research collections used under their licenses, and the email corpus is style-rewritten sample by sample before use. Private data stays inside a trusted boundary, evaluation on held-out real samples runs locally, and no private sample is sent to an external API. The guarantee is sample-level: under a per-person threat model, group privacy applies and the effective ε is weaker. A coherent fictional company is also a plausible phishing substrate, so we consider release of generated corpora, rather than of the pipeline, the decision that warrants review.

REPRODUCIBILITY STATEMENT

§3 states the mechanisms, sensitivities and composition and §3.4 the schema, release, constraint module, and validation agent. Every run writes a manifest carrying the achieved (ε, δ), σ and composition, with its configuration copied into the output directory before the run; embeddings, annotations and evaluation cards are content-addressed caches, so re-scoring is deterministic. The appendices give the per-algorithm and per-model numbers behind every ranking claim.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016.

Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning (ICML)*, 2018.

Preetha Chatterjee, Kostadin Damevski, Nicholas A. Kraft, and Lori Pollock. Software-related Slack chats with disentangled conversations. In *International Conference on Mining Software Repositories (MSR)*, pp. 588–592, 2020.

Eli Chien, Yuzheng Hu, Ryan McKenna, Shanshan Wu, Zheng Xu, and Peter Kairouz. MAPLE: Metadata augmented private language evolution. In *Conference on Language Modeling (COLM)*, 2026. arXiv:2603.19258.

Jinshuo Dong, Aaron Roth, and Weijie J. Su. Gaussian differential privacy. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):3–37, 2022.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. WorkArena: How capable are web agents at solving common knowledge work tasks? In *International Conference on Machine Learning (ICML)*, 2024.

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

EDRM. EDRM Enron email data set v2. Internet Archive, https://archive.org/details/edrm.enron.email.data.set.v2.xml, 2013. Custodian archives with native attachments and their extracted text.

Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*, 2024.

Google. Collaborate with Gemini in Gmail. https://support.google.com/mail/answer/14355636?hl=en, n.d. Accessed September 8, 2026.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Charlie Hou, Akshat Shrivastava, Hongyuan Zhan, Rylan Conway, Trang Le, Adithya Sagar, Giulia Fanti, and Daniel Lazar. PrE-Text: Training language models on private federated data in the age of LLMs. In *International Conference on Machine Learning (ICML)*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Alexey Kurakin, Natalia Ponomareva, Umar Syed, Liam MacDermed, and Andreas Terzis. Harnessing large-language models to generate private synthetic text. *arXiv preprint arXiv:2306.01684*, 2023.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations (ICLR)*, 2022.

Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, Harsha Nori, and Sergey Yekhanin. Differentially private synthetic data via foundation model APIs 1: Images. In *International Conference on Learning Representations (ICLR)*, 2024.

Zinan Lin, Tadas Baltrusaitis, and Sergey Yekhanin. Differentially private synthetic data via APIs 3: Using simulators instead of foundation models. *arXiv preprint arXiv:2502.05505*, 2025.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL*, pp. 11065–11082, 2024.

Justus Mattern, Zhijing Jin, Benjamin Weggenmann, Bernhard Schölkopf, and Mrinmaya Sachan. Differentially private language models for secure data sharing. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4860–4873, 2022.

Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the NIST contest: A scalable and general approach to differentially private synthetic data. *Journal of Privacy and Confidentiality*, 11(3), 2021.

Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. AIM: An adaptive and iterative mechanism for differentially private synthetic data. *Proceedings of the VLDB Endowment*, 15(11): 2599–2612, 2022.

Microsoft. What is Microsoft Copilot? https://learn.microsoft.com/en-us/microsoft-365/copilot/microsoft-365-copilot-overview, 2026. Accessed September 8, 2026.

Ilya Mironov. Rényi differential privacy. In *IEEE Computer Security Foundations Symposium (CSF)*, 2017.

Arindam Mitra, Luciano Del Corro, Guoqing Zheng, Shweti Mahajan, Dany Rouhana, Andres Codas, Yadong Lu, Wei-ge Chen, Olga Vrousgos, Corby Rosset, Fillipe Silva, Hamed Khanpour, Yash Lara, and Ahmed Awadallah. AgentInstruct: Toward generative teaching with agentic flows. *arXiv preprint arXiv:2407.03502*, 2024.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. Sentence-T5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics (ACL)*, pp. 1864–1874, 2022.

Douglas Oard, William Webber, David Kirsch, and Sergey Golitsynskiy. Avocado research email collection. Linguistic Data Consortium, LDC2015T03, 2015.

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

Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H. Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Guha Thakurta. How to DP-fy ML: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 77:1113–1201, 2023.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4), 2009.

Olly Styles, Sam Miller, Patricio Cerda-Mardini, Tanaya Guha, Victor Sanchez, and Bertie Vidgen. WorkBench: A benchmark dataset for agents in a realistic workplace setting. *arXiv preprint arXiv:2405.00823*, 2024.

Namjoon Suh, Ritam Dutt, Vivian Lai, Aynaz Taheri, Hyeonsu B. Kang, Anastasia Kuzminykh, and Alex C. Williams. Pike: Evidence-guided retrieval for email question answering, 2026. URL https://github.com/namjoonsuh/PIKE/blob/main/PIKE_manuscript.pdf. Unpublished manuscript.

Yidan Sun, Viktor Wang, Xin Huang, Bryan Hooi, and Bingsheng He. SynBench: A benchmark for differentially private text generation. *arXiv preprint arXiv:2509.14594*, 2025.

Xinyu Tang, Richard Shin, Huseyin A. Inan, Andre Manoel, Fatemehsadat Mireshghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. Privacy-preserving in-context learning with differentially private few-shot generation. In *International Conference on Learning Representations (ICLR)*, 2024.

Shuaiqi Wang, Vikas Raunak, Arturs Backurs, Victor Reis, Pei Zhou, Sihao Chen, Longqi Yang, Zinan Lin, Sergey Yekhanin, and Giulia Fanti. Struct-bench: A benchmark for differentially private structured text generation. *arXiv preprint arXiv:2509.10696*, 2025.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Tong Wu, Ashwinee Panda, Jiachen T. Wang, and Prateek Mittal. Privacy-preserving in-context learning for large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Chulin Xie, Zinan Lin, Arturs Backurs, Sivakanth Gopi, Da Yu, Huseyin A. Inan, Harsha Nori, Haotian Jiang, Huishuai Zhang, Yin Tat Lee, Bo Li, and Sergey Yekhanin. Differentially private synthetic data via foundation model APIs 2: Text. In *International Conference on Machine Learning (ICML)*, 2024.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned LLMs with nothing. In *International Conference on Learning Representations (ICLR)*, 2025.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A. Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. Differentially private fine-tuning of language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Da Yu, Peter Kairouz, Sewoong Oh, and Zheng Xu. Privacy-preserving instructions for aligning large language models. In *International Conference on Machine Learning (ICML)*, 2024.

Xiang Yue, Huseyin A. Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. Synthetic text generation with differential privacy: A simple and practical recipe. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

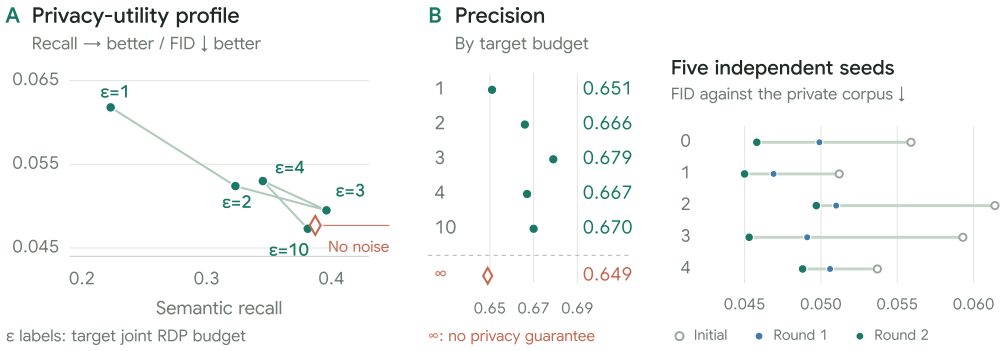
Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems*, 42(4), 2017.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 5905–5921, 2021.

A ADDITIONAL RESULTS

The additional results follow the main evaluation. **RQ1**: Tables 2 and 5 report the DP-SGD comparison and privacy-budget sweep. **RQ2**: Table 7 reports participant counts and sender-frequency divergences for the component ablation. **RQ3**: Table 3 gives the fine-tuning measurements. **RQ4**: Table 6 gives the retrieval rank-agreement statistics. **RQ5**: Table 4 compares the three further data types with their own same-distribution references.



(a) **(b)**

Figure 5: (a) Privacy budget sweep. The left panel plots recall against FID and the right panel reports precision. The hollow diamond is a no-noise reference ($\sigma = 0$), plotted separately from the private runs. Utility changes little above $\varepsilon \approx 3$, and $\varepsilon = 10$ approaches the reference (Table 5). **(b) FID decreases across five independent runs.** FID over the first two PE rounds in five independent runs, each drawing its own DP release and its own PE noise; the change is negative in every run, and negative again under three other canonicalizations of the same samples.

Table 5: Privacy budget sweep. Six budgets under one fixed configuration (2,000 private, 400 synthetic, 10 rounds, shared noise multiplier for the release and the vote, composed using Rényi differential privacy (RDP) accounting); each arm retains all 400 table rows and uses model-driven selection. The $\varepsilon = \infty$ arm is a no-noise utility reference with no privacy guarantee.

ε	σ	P ↑	R ↑	FID ↓
1	24.51	0.651	0.223	0.0618
2	12.50	0.666	0.323	0.0524
3	8.49	0.679	0.396	0.0495
4	6.48	0.667	0.345	0.0530
10	2.84	0.670	0.381	0.0473
∞	0 (no noise)	0.649	0.387	0.0477

The sweep uses one seed per budget and a fixed configuration distinct from the component ablation in Table 1. Comparisons are made within the sweep.

Table 6: Do the two corpora rank retrieval systems the same way? Eleven retrieval algorithms (BM25, dense, hybrid RRF, three LLM rerankers and five PIKE variants (Suh et al., 2026)) run over the real and the synthetic corpus, each on its own corpus-grounded benchmark built by one generator with one seed, at the same cutoffs; the two leaderboards are then compared. Rank agreement is positive on all three benchmarks; the final columns report cutoff sensitivity and adjacent-pair separation.

Benchmark	q	alg.	metric	ρ	τ	w-τ	r	top-1	ρ range	sep.
Email, 1 thread	50	11	`recall@3`	0.659	0.495	0.642	0.625	yes	-0.05–0.78	0/10
Email, 2 threads	25	11	`coverage@5`	0.618	0.523	0.522	0.789	undef.†	0.47–0.82	0/10
Documents	50	9	`recall@3`	0.898	0.806	0.791	0.878	undef.†	0.61–1.00	0/8

q is the number of queries per side, *alg.* the number of systems ranked, ρ *range* the span of the Spearman correlation over all available cutoffs, and *sep.* the number of adjacent leaderboard pairs whose gap exceeds $1.96\sqrt{2p(1-p)/n}$. †The real-side winner is tied, so top-1 agreement is decided by the sort order rather than by the data, and is reported as undefined.

Table 7: Participant counts and attribute fidelity. The same arms as Table 1, with additional corpus-level statistics. Round-10 values, email corpus.

Arm	synthetic participants		marginal JS vs private ↓			
	senders	H (bits)	sender	turns	msg-len	thread-len
Aug-PE	539	8.26	0.073	0.450	0.198	0.337
+ base schema	13	3.02	0.837	0.462	0.388	0.123
+ derived attrs., no module	569	8.41	0.091	0.394	0.276	0.233
Agentic-PE, random selection	79	3.74	0.081	0.284	0.162	0.160
Agentic-PE (full)	81	3.78	0.072	0.284	0.178	0.164
private reference	*748*	*6.69*	*0.000*	*0.000*	*0.000*	*0.000*

JS columns are Jensen–Shannon divergences of the synthetic marginal against the private one, so lower is closer. Participant counts are measured at round 10. The private corpus contains 748 senders in 2,000 threads; each synthetic corpus contains 400 threads.

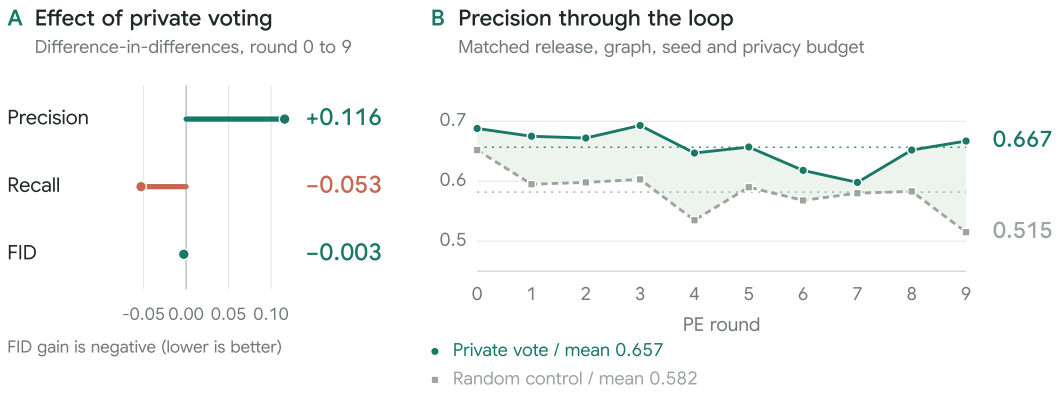


Figure 6: Private voting versus a matched random-selection control. *Left*: the difference between the arms' round-0-to-round-9 changes. *Right*: per-round semantic precision; dotted lines mark ten-round means. Precision falls in both arms but falls less with private voting. The control shares the release, module, variation mode, seed, and budgets.

Additional auto-reply fine-tuning. We evaluate the same models before and after LoRA fine-tuning on Agentic-PE data. Mean judge scores improve in all four model/benchmark combinations; three of the four 95% intervals include zero (Figure 7).

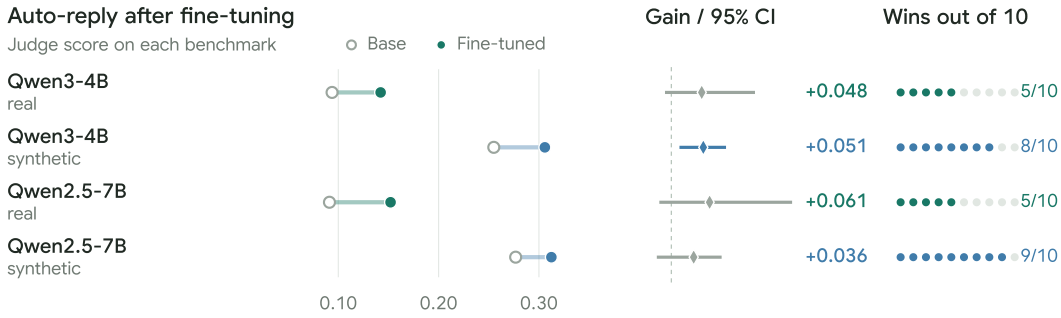


Figure 7: Additional auto-reply fine-tuning results. The same models before and after LoRA fine-tuning on Agentic-PE under the pre-registered judge; *wins* counts examples whose fine-tuned reply outscores the base reply. Mean scores rise in all four combinations, but three 95% intervals include zero.

B Implementation Details

Attribute schema. Each data type uses nine categorical attributes with fixed vocabularies and bucket boundaries. Semantic properties, such as intent or workflow stage, are annotated by a model inside the trusted boundary; structural properties are derived from measurements. The schema is small to limit sparsity at a fixed corpus size and privacy budget, and its categories are fixed independently of the private data (Appendix C). Per-sample attributes preserve sensitivity one because each sample still contributes to one joint attribute cell. Email's sender-frequency category additionally uses the DP frequency pre-release.

Attribute release. The annotated table contains one row per private sample and $k = 9$ columns. Tabular PE synthesizes P over T_r rounds (Lin et al., 2024). Sampling is stratified by the subject-matter attribute to preserve the released topical mix. We construct P once: later rounds change which candidates fill its rows while keeping the indexed rows fixed. This persistent table connects the initial generation conditions to the coverage constraints used in every selection round.

Constraint module and validation agent. The released attributes inform the roster size and the data-type requirements. The module creates fictional names, titles, departments, and relationships, then adds the relevant context rules (Table 8). Validation checks surface form, participant references, and temporal consistency, with model repair restricted to local failures. Discarded candidates are not resampled; a round's population is the number of table rows with at least one survivor. A configured loss tolerance prevents the loop from continuing with a substantially reduced population. Appendix F lists the vocabularies, release steps, thresholds, and repair settings; Appendix G describes the checks.

B.1 Evaluation Protocol

We first compare corpus quality with DP synthesis baselines (RQ1) and analyze the contributions of the components (RQ2). We then test the corpus as training data (RQ3) and as a benchmark for comparing systems (RQ4). Finally, we adapt the framework to three further data types (RQ5). Table 10 connects these questions to semantic, structural, and downstream measurements.

Corpora. Table 9 lists the four corpora. **Email is the primary experiment**: RQ1–RQ4 focus on email, with document retrieval as an additional evaluation benchmark. The other three data types test adaptation of corpus synthesis (RQ5). For the email corpus, 174 threads are held out of synthesis entirely and used only to evaluate downstream models on *real* data.

Metrics and comparisons. Semantic scores measure similarity in an embedding space; structural scores assess parsing and participant consistency. Shape JS is the mean Jensen–Shannon divergence, in bits, between synthetic and private distributions of the derived categorical attributes. For email

Table 8: Shared context and validation rules by data type. Each adapter supplies data-type-specific context and checks while reusing the evolution loop and privacy accounting.

Data type	Adapter component	What this type supplies
Email	*a sample is*	one email thread
	shared context	recipient and broadcast rules over the organization
	validation checks	headers, sender on roster, stable subject, monotone dates
Chat	*a sample is*	one conversation in a channel
	shared context	channels with a purpose, membership and bots
	validation checks	every participant speaks, released turn and length shape
Meetings	*a sample is*	one agenda item
	shared context	committees with scope, chair, members and cadence
	validation checks	every attendee speaks, chair on register, timestamps within the meeting window
Documents	*a sample is*	one internal document
	shared context	who may author a type and who must approve it
	validation checks	section count in range, approver present, length capped

these are the sender-frequency category and the average-message-length category, weighted equally. This differs from the individual sender-count and message-length histograms in the population analysis.

Semantic estimates use matched sample sizes. Where we normalize by a same-distribution reference, it is computed from disjoint subsets of the private corpus at those sizes; a ratio of 1.0 equals that reference score. Privacy budgets are matched separately: removing a mechanism changes the composition, so we solve for each arm's noise multiplier σ rather than copy it from the full method. Unless stated, $\varepsilon = 5.2985$, $\delta = 10^{-5}$, $m = 400$ samples over $T = 10$ rounds (Appendix C). The DP-SGD baseline uses $\sigma = 1.3818$, a clipping bound $C = 1.0$, and 100 Poisson-subsampled steps, giving $\varepsilon \leq 5.293$ at the same δ.

Table 9: The four corpora. All are public research collections used under their licenses; the email corpus is style-rewritten sample by sample inside the trusted boundary before use (Appendix C), and the other three are prepared by the deterministic procedure of Appendix E.

Corpus	Source	Samples	A sample is	Role
Email	Avocado (Oard et al., 2015)	2,000	one email thread	**primary** (RQ1–RQ4)
Chat	Slack (Chatterjee et al., 2020)	2,000	one conversation	transfer (RQ5)
Meetings	QMSum (Zhong et al., 2021)	232	one agenda item	transfer (RQ5)
Documents	Enron (EDRM, 2013)	284	one document	transfer (RQ5), retrieval

Table 10: The three families of measurement. Semantic scores are computed on a canonical topic summary of each sample, embedded with a sentence encoder (Ni et al., 2022).

Family	Measure	What it answers	
(i) Semantic fidelity: synthetic against private, in embedding space			
	precision	Are synthetic samples the kind of sample the private corpus contains? (Kynkäänniemi et al., 2019)	↑
	recall	How much of the private corpus's semantic space do they cover?	↑
	FID	How far apart are the two distributions overall? (Heusel et al., 2017)	↓
(ii) Structural validity and consistency: individual samples and the collection			
	validity	Share of samples passing the deterministic checks of §3.4	↑
	shape JS	Mean divergence of derived attribute distributions; email uses sender-frequency and message-length categories	↓
(iii) Downstream utility: training and evaluation on a substitute corpus			
	NLL, ROUGE-L	Negative log-likelihood (NLL) and longest-common-subsequence overlap with reference replies (ROUGE-L), evaluated on held-out *real* samples after fine-tuning on synthetic data	↓, ↑
	judge score	Rating of a model's generated replies by a pre-registered large language model (LLM) judge (Zheng et al., 2023)	↑
	Spearman ρ	Agreement between a leaderboard computed on the synthetic corpus and on the real one	↑

C CORPORA AND ESTIMATORS

Corpora. The email corpus is deduplicated and rewritten sample by sample into modern corporate English by an independent model that preserves entities, dates and numbers; the rewrite is a per-sample map inside the trusted boundary, so the DP sample structure is unchanged. The meeting corpus uses one agenda-item sample per meeting. The document corpus is built from the extracted text of *attachments* in the EDRM Enron v2 release (EDRM, 2013), not from the `.eml` messages that share the same custodian archives, so it stays disjoint from the email modality it is meant to contrast with.

Provenance of the schema. The field names, the categorical vocabularies, and the boundaries of the two derived buckets are constants of the implementation: they are written from generic business-communication categories, are not tuned to any corpus, and are identical for every arm and for any pair of neighboring datasets, so the release mechanism is a fixed function of the data rather than a data-dependent one. Concretely, for email the length bucket splits a thread's mean message at 350, 650 and 1,100 characters and the sender tier splits a sender's frequency at 1, 4 and 15 samples; neither boundary is a quantile of the corpus. A schema fixed in this way satisfies the data-independence assumption in Proposition 1.

One coupling in the derived fields, and how it is paid for. `sender_tier` buckets a sample by how often its primary sender occurs in the corpus, so it is a function of the dataset rather than of the sample alone: computed exactly, adding one sample can move every other sample of that sender across a tier boundary, and the joint histogram would not have sensitivity one. The pipeline therefore releases the sender frequencies first. The vector of per-sender sample counts changes in one coordinate by one between neighboring datasets, so publishing it with Gaussian noise at the shared σ is a mechanism of $\Delta_2 = 1$; the tier is then a fixed threshold of the noisy count. Adaptive composition then uses sensitivity one for the second stage conditional on each fixed output of the first.

Lemma 1 (Conditional sensitivity of the attribute release)**.** *Let M_{freq} output a noisy count vector y indexed by a sender universe fixed independently of D, and for a fixed threshold rule let $\tau_y(u)$ be the tier of sender u under y. For a dataset D let $H(D; y)$ be the joint histogram over $\mathcal{A}$ in which each sample $r \in D$ contributes one count to the cell given by its eight per-sample attributes together with $\tau_y(\text{sender}(r))$. Then for every fixed y and every pair D, D' differing in the addition or removal of one sample, $\|H(D; y) - H(D'; y)\|_2 = 1$.*

Proof. Write $D' = D \cup \{r_0\}$. Fix y. For any $r \in D$, the cell of r is determined by its own eight attributes and by $\tau_y(\text{sender}(r))$. Both are functions of r alone once y is fixed, because y does not depend on which of D, D' is being processed. Hence every $r \in D$ occupies the same cell in $H(D; y)$ and in $H(D'; y)$, and the two histograms differ only by the single count contributed by r_0. □

Lemma 1 gives conditional sensitivity one for the attribute query. Proposition 1 composes it with the frequency release, using $T_f = 1$. The other eight fields are functions of individual samples.

The sender universe. The frequency vector is indexed by a sender universe fixed independently of the samples, with i.i.d. noise per coordinate; the implementation draws that noise only for the senders that occur, which is the lazy realization of the same mechanism and is why an unseen sender is harmless. The draws are i.i.d., so which coordinate receives which draw does not change the joint law, and the realized subset has the distribution the fixed-universe mechanism induces on it, with the fresh randomness a deployment uses rather than the fixed seed that makes our runs reproducible. Take neighbors differing in one sample r. If r's sender already occurs, one coordinate of the count vector moves by one, which is what the pre-release pays for, and other samples of that sender may cross a tier boundary. If r's sender is new, its coordinate is read by r alone: every other sample keeps the tier it had, the two joint histograms differ by one row, and $\Delta_2 H = 1$ is already paid by the attribute release. The support itself is never published: sender identifiers do not leave the trusted boundary, and the released objects are the noisy histogram over the fixed attribute domain, the synthetic attribute rows drawn from it, and the noisy votes.

Budget-matched Aug-PE. We calibrate Aug-PE's voting noise so that it uses $\varepsilon = 5.2985$, matching the full method's combined attribute and voting account.

Randomness. The reported runs draw every Gaussian from a seeded generator so that a run reproduces exactly. A deployment must draw fresh randomness: a known seed voids the guarantee for the vote as much as for the release.

Estimators. Scores are computed at matched sample sizes because the k-NN estimator reads each set's radius off its own k-th neighbor, so an unmatched comparison rewards the larger set. A raw FID is not comparable across corpora, which is why every cross-corpus statement in RQ5 is a ratio against the same-distribution floor measured on two disjoint halves of the private corpus at the same sizes.

D THEORETICAL ANALYSIS

This appendix follows the analysis that accompanies PE: Lin et al. (2024) prove convergence of the loop under explicit modeling assumptions (their Appendices E and F), and the simulator instantiation argues which of those assumptions its changes touch (Lin et al., 2025). We do the same for Agentic-PE. The privacy bound below accounts for the composed Gaussian steps. The convergence statements use the Gaussian embedding-space variation model of Lin et al. (2024), with assumptions (M1)–(M6) stated below.

D.1 ACCOUNTING FOR THE GAUSSIAN STEPS

Agentic-PE composes $N = T_f + T_r + T$ Gaussian mechanisms of sensitivity one at a shared noise multiplier σ: the frequency pre-release, the attribute release and the voting rounds (Proposition 1; Lemma 1 supplies the conditional sensitivity that lets the attribute release count as sensitivity one). Every other stage of Algorithm 1 reads DP outputs and synthetic samples only. The implementation uses Rényi DP to account for these Gaussian steps. The Rényi curve of one such mechanism is $\varepsilon(\alpha) = \alpha/(2\sigma^2)$, so adaptive composition gives $\varepsilon_N(\alpha) = N\alpha/(2\sigma^2)$, which is the curve of a single Gaussian mechanism with effective multiplier $\sigma/\sqrt{N}$ (Mironov, 2017; Dong et al., 2022). Converting this curve gives the bound

$$\varepsilon = \min_{\alpha > 1} \frac{N\alpha}{2\sigma^2} + \frac{\log(1/\delta)}{\alpha - 1},$$

for any $\delta \in (0, 1)$. The implementation minimizes the same expression over a fixed grid of orders, giving a conservative bound, and solves for σ at the target (ε, δ). Each run records the resulting bound, σ and step count in its manifest. Agent operations are post-processing under Proposition 1.

D.2 What the attribute channel carries

The attribute release provides explicit targets over attribute cells and their masses (§3.3), rather than a preference over text candidates. The attribute domain is finite, with K cells, so the convenient model of the tabular release is a single thresholded voting round over a candidate set that contains every cell; on a finite domain the covering condition that Lin et al. (2024) obtain from exponentially many candidates holds by enumeration. Each private sample votes for its own cell, each count receives $\mathcal{N}(0, \sigma^2)$ noise, cells whose noisy count clears a threshold H enter the release, and output table rows are allocated to released cells in proportion to their noisy counts. The implementation spends T_r steps of the tabular synthesizer on this channel rather than one; Proposition 1 accounts for the steps actually taken, and the one-round model is an analysis simplification of the same kind as the variation model below.

Theorem 1 (Release utility)**.** *Let n_k be the number of private samples in cell k, let $\tau \in (0, 1)$, and set $H = \sigma\sqrt{2\ln(2K/\tau)}$. With probability at least $1 - \tau$, simultaneously: (i) every cell with $n_k \geq 2H$ is released; (ii) no empty cell is released; and (iii) every released count is within H of its private count.*

Proof. Each of the K noise draws exceeds H in absolute value with probability at most $2\exp(-H^2/2\sigma^2) = \tau/K$, so a union bound puts all K draws in $[-H, H]$ with probability at least $1 - \tau$. On that event a cell with $n_k \geq 2H$ has noisy count at least H and is released, an empty cell has noisy count at most H and is not, and every released count is within H of its private count. □

For this simplified release model, attribute cells with mass at least $2H$ are retained on the stated high-probability event. This is a sufficient condition for retention. Part (iii) bounds released count errors; allocating and sampling table rows then determines the realized marginal distributions measured in §4. Theorem 2 imposes a related lower-mass assumption for its convergence result.

D.3 Convergence of the constrained loop

For PE, Lin et al. (2024) prove: without noise, the loop brings every private point within η of a surviving candidate, in Wasserstein distance $W_p \leq \eta$ for every $p \in [1, \infty]$, after $T \gg d\log(D/\eta)/\log L + \log(n/\tau)$ rounds, where d is the embedding dimension, D the diameter of the support and L the number of variations per survivor; with noise, the same bound holds when every private point has multiplicity $B \gg \sigma\sqrt{\log(TLn/\tau)}$, because the noisy selection then coincides with the noiseless one with high probability. Three of the four changes Agentic-PE makes map onto that analysis without new argument, the same way the simulator instantiation's changes do (Lin et al., 2025): attribute-conditioned initialization changes the initial support (their D), not the procedure; conditioning the variation call on the constraint module changes what the samples say, not the Gaussian model of how they spread; and the validation agent is covered by assumption (M5) below. The substantive change is selection: a constrained per-row argmax replaces thresholding, and the rest of this subsection handles it.

The modeling assumptions, matching Lin et al. (2024) where the algorithms coincide:

- (M1) Embeddings of private samples and candidates lie in an ℓ_2 ball of diameter D, and distances are ℓ_2 in embedding space.
- (M2) $\mathrm{Var}(z)$ returns z together with L samples of $\mathcal{N}(\phi(z), \sigma_i^2 I_d)$ at every scale $\sigma_i = D\sqrt{\log L}/(2^i d)$ for $1 \leq i \leq r = \log(D/\eta)$. The candidate pool for each table row has size $L' = Lr + 1$. The implementation uses $b = 2$ variations without retaining the parent.
- (M3) The analyzed output is the vote distribution over the final survivors, the convention of the non-private analysis of Lin et al. (2024); the noisy run reduces to it on the survival event of Lemma 2.

(M4) Multiplicity, per cell: the occupied attribute cells partition D into clusters, and all n_k samples of cell k share one embedding. This is the multiplicity assumption of Lin et al. (2024) with one cluster per cell; their discussion of why multiplicity is the right simplification applies unchanged, with a cluster read as a topic that recurs n_k times.

(M5) Alignment and admissibility: each cluster's nearest candidate was generated for a table row that matches the cluster's cell, candidates for that row receive votes only from that cell, and neither the validation agent nor the admissibility constraints remove a cluster's nearest candidate. The first two parts are what attribute-conditioned generation and attribute-matching checks are for; the third is an oracle assumption about the validation agent, and the closing remarks say where the implementation can violate it.

(M6) Coverage: every occupied cell is assigned at least one table row. This is an additional allocation assumption. Theorem 1 bounds cell retention in the simplified release; finite sampling of m table rows does not by itself ensure a row for every retained cell.

Proposition 2 (Row coverage)**.** *Under the selection rule of Eq. 1, every table row is represented by exactly one survivor in every round in which its admissible set is non-empty, for any votes, any noise and any candidate quality.*

Proof. Immediate from the definition: the survivor set is $\{c_1^\star, \ldots, c_m^\star\}$ with $c_i^\star \in A_i \subseteq C_t^{(i)}$, where every candidate in $C_t^{(i)}$ carries row index i by construction. The proposition concerns these indices. □

In the selection ablation, greedy selection retains 20% of the released row indices by round ten (Figure 2a).

Lemma 2 (Candidate survival under noise)**.** *Fix a round and an occupied cell k whose nearest candidate $z^\star$ belongs to the candidate pool for row i. Under (M4) and (M5), if $n_k \geq 2\sigma\sqrt{\ln(L'mT/\tau)}$, then $c_i^\star = z^\star$ with probability at least $1 - \tau/(mT)$.*

Proof. By (M4) and (M5) the cluster's n_k votes all land on $z^\star$, every other candidate for that row has zero votes, and $z^\star$ is admissible. The noisy argmax prefers a competitor only if that competitor's noise exceeds $z^\star$'s by at least n_k; each such difference is $\mathcal{N}(0, 2\sigma^2)$, so the probability is at most $L' \exp(-n_k^2/4\sigma^2) \leq \tau/(mT)$. □

Theorem 2 (Convergence of the constrained loop)**.** *Assume (M1) through (M6), $\log L \ll d$, and $n_k \geq 2\sigma\sqrt{\ln(L'mT/\tau)}$ for every occupied cell k. With probability at least $1 - 2\tau$, Agentic-PE run for T rounds outputs a vote distribution P_T with $W_p(D, P_T) \leq \eta$ for every $p \in [1, \infty]$, whenever*

$$T \gg \frac{d \log(D/\eta)}{\log L} + \log(n/\tau).$$

Proof. Under (M6) there are at most m occupied cells, so by Lemma 2 and a union bound over cells and rounds, with probability at least $1 - \tau$ every cluster's nearest candidate survives every round: the noisy constrained selection coincides with the noiseless rule that keeps each cluster's nearest candidate, which is the reduction used in the proof of Theorem 2 of Lin et al. (2024). Conditioned on that event, their contraction argument applies per cluster without change. Whenever the cluster distance $s = \|\phi(x_k) - \phi(z^\star)\|_2$ exceeds η, some scale σ_i satisfies $\sigma_i d/\sqrt{\log L} \in [s/2, s]$, and one of the L variations at that scale contracts the distance by a factor $1 - \log L/(4d)$ with probability at least $1/2$ (their Claim 1); the parent is retained under (M2), so the distance never grows. A Chernoff bound gives contraction in at least $T/4$ rounds with probability $1 - \exp(-\Omega(T))$, and a union bound over the at most n clusters costs the $\log(n/\tau)$ term. After T as stated, every cluster is within η of its surviving candidate. For the Wasserstein bound, transport each cell's mass n_k/n to that cell's surviving candidate: under (M3) the vote distribution places exactly n_k/n there, every unit of mass moves a distance at most η, and $W_p \leq \eta$ follows for every p. □

Connection to the implementation. The convergence theorem uses Gaussian variations at multiple scales, retained parents, and the alignment and oracle-admissibility conditions in (M5). The implementation uses two language-model variations per survivor without retained parents. Its participant cap and interactive-mix band can exclude a cluster's nearest candidate; the loop then keeps

the highest-voted admissible alternative. The utility bounds provide sufficient mass conditions of order $\sigma\sqrt{\log(\cdot)}$. Privacy depends on the DP-access conditions in Proposition 1, independently of these utility assumptions.

E PREPARATION OF THE CHAT, MEETING AND DOCUMENT CORPORA

Preparation runs inside the trusted boundary and is not itself released. The stated guarantee applies to the prepared corpus: neighboring corpora differ in one prepared sample (§3). A guarantee over raw source records additionally requires adjacency-preserving preprocessing. We use one agenda item per meeting and collapse document revisions to define the units. Preparation is deterministic given one seed. Anonymization is a fixed function of the seed and the raw identifier, so a person keeps a single fictional identity across every sample they appear in, and the correspondence itself is never stored. Real wall-clock time is discarded, and each sample is placed on a synthetic timeline of its data type. Private samples are rendered by the same deterministic renderer the generator uses, so the semantic scores of §4 compare content under a common format. Participant and community identifiers are replaced while subject matter, such as programming-language names, is retained. After each build, the output is searched for source identifiers and residual hits are checked against replacement names and retained subject vocabulary. The email corpus, which is model-rewritten rather than rule-scrubbed, is covered in Appendix C.

Chat: one sample is one conversation. The source (Chatterjee et al., 2020) distributes workspace chat already disentangled into conversations by human and semi-automatic annotation, so the conversation boundary is upstream annotation rather than a choice made here. A conversation longer than forty messages is cut back to the last change of speaker before the limit, so a sample does not end mid-exchange; conversations with fewer than three messages or a single speaker, conversations that are mostly code (over 60% of characters inside code markup), and mostly non-ASCII conversations are dropped. The language filter is deliberately lenient because chat is terse and full of identifiers, and an aggressive one would bias the corpus toward prose. From roughly 39,000 raw and 23,000 eligible conversations, 2,000 are sampled, stratified over source channels in proportion to eligible volume with no channel permitted more than half of the corpus. Anonymization runs before rendering, with two properties the conversational structure forces. The pseudonym map is primed over every identifier in the eligible pool rather than per conversation, because a message routinely names somebody the disentanglement placed in a different conversation; and all substitutions happen in a single pass over one combined pattern, so a later rule cannot rewrite an earlier rule's replacement. Usernames become fictional names and channels fictional channels; a mention of somebody in the conversation is rewritten to their pseudonym, and a mention of somebody absent is demoted to a bare name, since the sample format treats an addressed absentee as an error. Addresses, URLs and IP addresses become fixed placeholders. Typographic punctuation is folded to ASCII: the source is full of curly quotes and the generator writes plain ASCII, a systematic character-level difference that is formatting, not content. Of the original timestamps, only the gaps between consecutive messages survive, clipped at four hours.

Meetings: one sample is one agenda item, and one meeting contributes one sample. The build keeps at most one segment per source meeting, which makes "remove one meeting" and "remove one sample" the same event; every meeting of the source (Zhong et al., 2021) survives the filters, giving the 232 samples of Table 9. The transcripts arrive already split into speaker turns, so no sample needs heuristic turn-splitting. Two artifacts of the release are corrected first: the same meeting ships several times, under its data split and again in aggregate directories, and is kept once; and transcription markup (vocal-sound and disfluency markers) and tokenizer spacing are removed, since the generator writes ordinary prose and the difference is formatting rather than content. Each session is then cut into agenda-item-sized windows of variable width between 3 and 30 turns. Windows with fewer than three turns, with under 400 or over 12,000 characters, or with mostly non-English text are dropped. Because one released attribute is the initiator's role, the role assignment is stated: a speaker whose name carries a presiding title chairs, and otherwise the most prolific speaker does, which is how a presiding officer behaves in a transcript; speakers whose names carry staff titles are staff; everyone else is a member. The outcome block is read off the closing turns rather than invented: the first disposition phrase ("motion carries", "approved as revised") becomes the decision, no match leaves the decision explicitly empty (an informational item must not acquire a fabricated approval), and sentences assigning follow-up work become action items, at most three. Speakers

and committee names map to fixed fictional pools, and meetings are placed on a synthetic weekday-morning cadence.

Documents: one sample is one document, after collapsing revisions. A business document rarely exists once: the same memo circulates as drafts and a final, and each revision would otherwise enter the corpus as a highly correlated row of its own, the same threat to adjacency that repeated meeting segments pose. Exact duplicates are removed by content hash, and near-duplicates are clustered on character 5-gram shingles (Jaccard similarity above 0.7), each cluster collapsed to its longest member, so "remove one document" removes one row. A document enters the pool only as prose: extracted attachment text of 400 to 20,000 characters, cut back to 6,000 at a paragraph boundary when longer, with alphabetic characters above 70% of non-whitespace, table-like lines under 20%, and English text; spreadsheets are excluded by file type. Each document is assigned one of the released type values by a fixed precedence of signals (the filename, then a memo-style header block, then the opening lines, defaulting to a memo), because body prose is the weakest signal: a memo that reviews figures is not thereby a report. Sampling caps any single type at 40% of the corpus, and when a cap binds the corpus comes up short rather than being backfilled from the dominant type. The prose is split into sections at heading-like lines, with the section count bounded by the type (one to three for a memo, one for an announcement, two to eight for the committing types), so the released type controls the structure of the sample rather than decorating its header. Anonymization replaces a fixed list of source-era people and organizations, the person and organization spans found by a named-entity recognizer, and capitalized name patterns, all with draws from fictional pools; each document receives a fictional author and, for the committing types, an approver; addresses, URLs and phone numbers become placeholders.

F Implementation parameters

Schema. Each schema has $k = 9$ columns, combining annotated attributes with vocabularies of three to ten values and derived attributes with three or four buckets. For email the annotated attributes are topic family, intent, workflow stage, urgency, artifact type, calendar linkage and communication mode, and the derived ones are a length bucket and a sender tier; the other three data types use the analogous sets summarized in Table 8.

Release. $T_r = 15$ Gaussian steps of the tabular synthesizer at noise multiplier σ, stratified draw of $m = 400$ attribute rows for email ($m = 60$ for transfer) on the subject-matter attribute, preceded by $T_f = 1$ Gaussian step at the same σ for the sender-frequency release the email sender tier is thresholded from (Appendix C); the other three data types read every attribute off the sample itself and run at $T_f = 0$. The whole account is therefore $T_f + T_r + T = 26$ compositions for email; the transfer corpora also run a shorter loop ($T = 3$), so their account is $T_r + T = 18$.

Loop. $T = 10$ rounds for email and $T = 3$ for the transfer corpora; variation fold $b = 2$ and parents not carried over, so $|C_t| = 800$ before validation on email; embeddings from a sentence encoder (Ni et al., 2022), and the vote is a nearest-neighbor histogram over them.

Validation agent. At most two repair attempts per sample, and only for surface failures; a batch aborts if more than $\max(3, 2\%)$ of it fails on content after repair. Rate-limited generation calls are retried rather than counted as failures.

Constrained selection. One survivor per table row, τ_u at one tenth of the population for any single participant, τ_ρ at five percentage points. The selection agent proposes a survivor set, and every proposal is verified against the constraints before it is accepted. If the agent call fails at runtime or returns an infeasible set, a deterministic fallback maximizes vote mass under the same constraints. Both implementations enforce the objective and feasible set used in the row-coverage comparison of RQ2.

G Generation-time quality control

§3.2 describes the validation agent; this section follows one drafted sample from the generation call to the vote and says what each step may change. Everything below reads the attribute rows, the constraint module and the drafts themselves, so Proposition 1 covers it as post-processing.

Structured drafting. Every generation and variation call returns a typed object conforming to the data type's schema. A reply that violates the schema is retried with the validator's complaint appended; a truncated reply is retried with an instruction to shorten prose rather than omit fields; a generation request that fails all retries is dropped and counted. Text is produced from the accepted object by the deterministic renderer of Appendix E, so formatting cannot vary between samples or between the private and synthetic sides.

Participant assignments are enforced after generation. The prompt asks for the roster the constraint module sampled for the attribute row, and the accepted object's participant list is then overwritten with exactly that roster; a sender index outside the roster is mapped back into it, and an attribute row calling for a broadcast is forced down to a single message from its designated sender.

Attribute matching. Each accepted object is read back through the same annotation that bucketed the private corpus and compared with its attribute row on the attributes a single sample determines. A miss names the field, the target and the produced value, and the sample is regenerated at most twice with that correction appended; a replacement is kept if it misses strictly fewer fields. Samples that still miss are kept and the residual is recorded, because dropping them would bias the corpus toward the shapes the generator finds easy. The selection rule of §3.2 then prefers, among candidates for the same table row, those that match its attributes.

The consistency audit. A second model call reads the draft together with three things the loop already holds: the module profiles of its participants, the earlier samples most related to it, and a ledger of facts established by samples already accepted. It returns the (possibly corrected) sample plus an audit of its checkable content: the calculations, commitment dates, signatures and stable facts the sample contains. Deterministic code then validates that audit: arithmetic is recomputed (nothing the model emits is executed), dates are checked against the sample's position on its timeline, titles against the module, stable facts against the ledger. Heuristics also flag a sample that mentions a computation, a deadline, a signature or a policy but audits none of them; each data type disables the inapplicable heuristics, such as signature checks for chat and spoken meetings. Failures are returned to the model by name for the next attempt, and the corrected version carries forward rather than restarting from the original draft. Three corrections bypass the model entirely, because the loop already holds the authoritative value: a participant's title is set from the module, an established fact keeps its established value, and a stated total that disagrees with its terms is recomputed and replaced in the prose. Repair is bounded by invariants: it may not change the participants, the number of messages, turns or sections, or the attribution of any of them to a speaker, and a response that changes these fields is rejected.

Failure handling. A sample that cannot be repaired within the attempt budget of Appendix F is dropped with its reason recorded; a round whose content failures exceed the tolerance aborts (§3.2). Throttling by the serving infrastructure is retried and excluded from that tolerance, since a rate-limited sample was never judged on its merits.

Facts accumulate across samples. The audited facts of every accepted sample enter the ledger the next audits read, so figures, policies and commitments stay consistent across samples generated rounds apart. The module and the ledger split the work of making the corpus one environment: the roster makes the same people recur, and the ledger keeps what they have already said consistent.

Selection re-checks the rendered text. Independently of the validation agent, which validates objects at drafting time, the selection stage re-reads every candidate from its rendered text alone, with deterministic checks per data type: an identity absent from the roster, a mention of somebody not in the room, a broadcast source that receives replies, body years ahead of the sample's date, a decision no discussion supports, a committing document without its approver. Candidates carrying a defect are removed from the admissible sets of Eq. 1, which is the defect clause of §3.2.

H RETRIEVAL: SYSTEMS AND PER-ALGORITHM SCORES

Benchmark construction. Each corpus has its own queries and gold sets, built by the same procedure. Let C be either corpus, of the same size on both sides. Draw n seed samples $x_1, \ldots, x_n$ from C with one fixed seed, and set

$$q_i = \mathrm{QueryGen}(x_i), \qquad G_i = \{x_i\} \cup \{x \in C : \cos(\phi(x), \phi(x_i)) \geq 0.90\},$$

with the same prompt, the same model, the same embedding ϕ, the same threshold and the same n on both sides. We compare the system rankings induced by $R^{\text{real}}(s)$ and $R^{\text{syn}}(s)$ using Spearman ρ. The real side reads private samples to write q_i and to compute G_i; that happens inside the same trusted boundary as the rest of the evaluation and nothing derived from it enters the released corpus or the privacy account.

Common setup, email benchmarks. Both email benchmarks index whole threads, 400 samples per side by construction, with no chunking or message-level splitting, and every system returns its top 15 threads. Dense retrieval embeds the first 1,500 characters of each thread with `text-embedding-ada-002` (1,536 dimensions); vectors are L_2-normalized so cosine equals dot product. Lexical retrieval is BM25 (Okapi, $k_1 = 1.5$, $b = 0.75$) over lower-cased alphanumeric tokens with no stop-word removal. Hybrid fusion is reciprocal-rank fusion with $k = 60$ over the full BM25 and dense rankings. The reranking and agent LLM is GPT-5-mini throughout, with no sampling temperature set. Queries are LLM-generated under a hard prompt that forbids reusing the seed's distinctive words, figures and identifiers; $n = 50$ seeds and seed 0 everywhere.

The eleven systems. Three single-stage baselines (`bm25_only`, `dense_cosine_only`, `hybrid_rrf_only`) score the whole corpus and return the top 15 directly. Three reranker systems (`bm25_llm_rerank`, `dense_llm_rerank`, `hybrid_llm_rerank`) take the top 30 from their first stage and issue one listwise LLM call showing each candidate's subject line and first 420 characters; the model must return a complete ordering as JSON, and any candidate it omits or invents keeps its first-stage position. The five agentic variants (the PIKE family Suh et al. (2026) in Figure 4) run a pipeline over an offline graph of person, topic and thread nodes: (i) one query-analysis call extracts a rewritten query, the people involved, topic keywords and a time window; (ii) five retrieval axes are formed (hybrid content retrieval at depth k_c, person match, topic match, time window, and broadcast threads), and the candidate set is the union of all pairwise axis intersections plus the top 15 of the content axis; (iii) candidates are reranked; (iv) a sufficiency critic either accepts the set or emits a requery, which re-runs retrieval and admits k_r further threads per round; (v) a final rerank returns the top 15. The system name encodes the configuration. `cos_`$\star$ reranks by embedding cosine; `llm_`$\star$ scores each candidate pointwise with an LLM over a top-40 pool. `checklist` critics first generate a two-to-three-item information-needs checklist and judge coverage against it; `cot` critics use a single chain-of-thought judgment. `d2`/`d4`/`d6` is the total round budget (at most 1, 3 and 5 requeries), and `shallow`/`deep` also scale the first-stage depth k_c (40/75/150) and the per-round admissions k_r (5/10/15).

Two-email benchmark. The same eleven systems and the same corpora. The 25 queries per corpus are generated from thread *pairs* within that corpus, with pairwise cosine in $[0.65, 0.88]$, related but not near-duplicates, and a generation is kept only if the generator certifies that answering requires both threads. The gold set is exactly the pair, and coverage@k is the fraction of the pair retrieved.

Documents benchmark. The document corpus uses a separate, modality-agnostic implementation with nine systems and no agentic variants: BM25, dense and hybrid first stages, each either returned directly or reranked by cosine or by a listwise LLM call over a pool of 12 candidates (24 for `hybrid_llm_deep`). Three differences matter when reading across benchmarks: its BM25 is a separate implementation (no IDF floor, minimum token length three); generated queries are rejected and regenerated when their IDF-weighted lexical overlap with the target sample exceeds 0.25; and its recall@k is a per-query hit rate (whether any gold sample appears in the top k) rather than the fraction of the gold set retrieved, so recall columns are not comparable across the two implementations. Both sides are cut to the smaller corpus size, 283 samples.

Figure 8 gives every system's score on both corpora for all three benchmarks, ordered by real-corpus score.

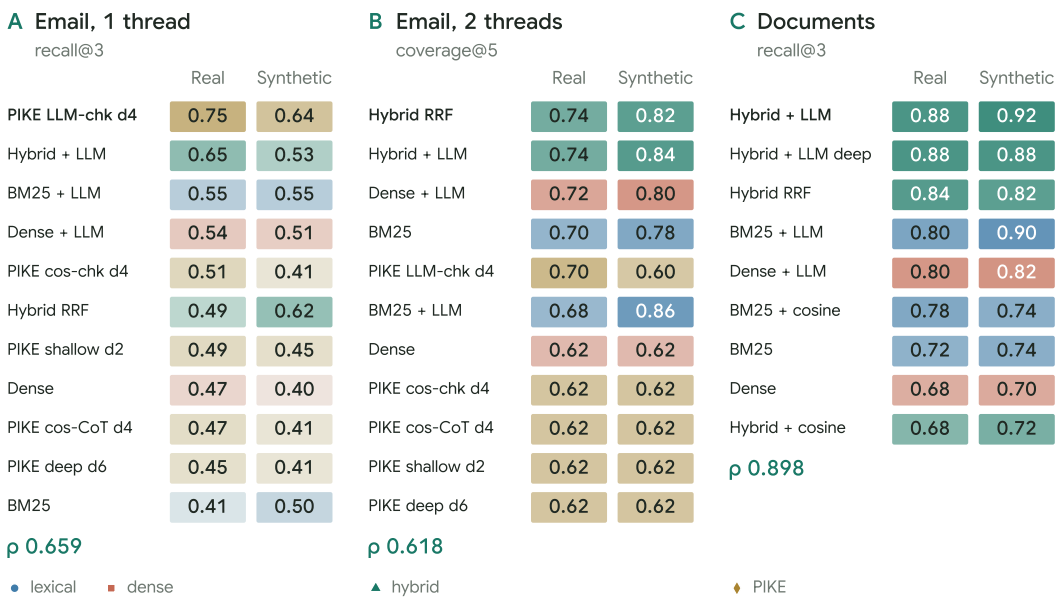


Figure 8: Per-system retrieval scores on the three benchmarks. One row per system, with paired columns for real-corpus and Agentic-PE-corpus scores, ordered by real-corpus score. Each cell reports the score; color marks the retriever family. The rank-agreement statistics for the same data are in Table 6.

I AUTO-REPLY: PER-MODEL AND PER-JUDGE SCORES

We compare ten candidate LLMs on ten real and ten synthetic thread prefixes per corpus. Under the pre-registered judge, we obtain Spearman $\rho = 0.733$ with a 95% bootstrap interval of $[0.006, 0.843]$ and identify the same top-ranked model on both benchmarks. Every model receives a higher mean score on synthetic examples, with increases of 0.34–1.04 (Figure 9). We therefore evaluate agreement in relative model rankings separately from the absolute score levels.

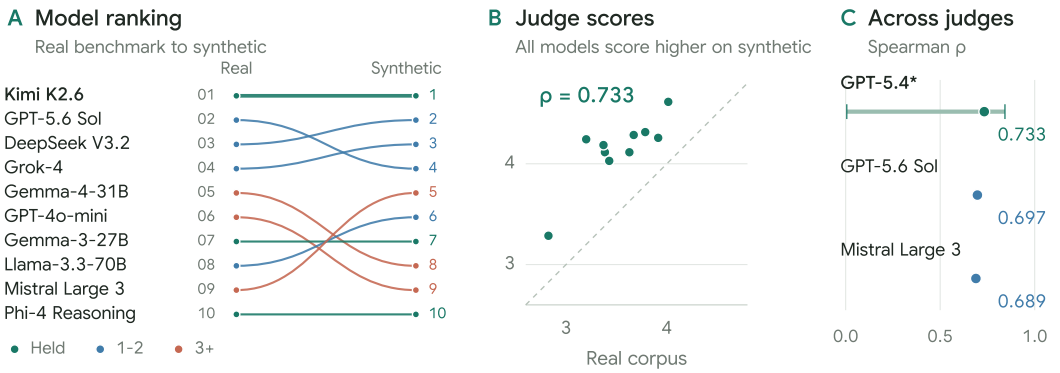


Figure 9: RQ4, reply models. *Left*: rank flow from the real leaderboard to the synthetic one; the winner is the same. *Middle*: the same ten models' judge scores, real against synthetic; the ranking correlates ($\rho = 0.733$) while every model scores higher on synthetic ($+0.34$ to $+1.04$), showing rank agreement alongside a difference in difficulty. *Right*: the same generations re-scored by three independent judges, which agree (Appendix I).

Figure 10 gives each candidate model's judged score and critical-error rate on both benchmarks under the pre-registered judge, and Figure 11 the rank-agreement statistics under each judge that completed.

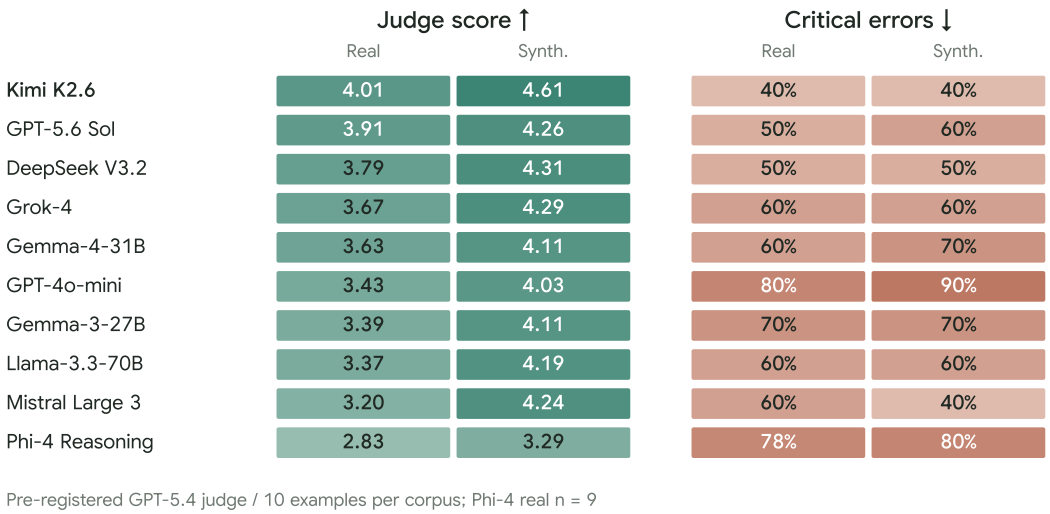


Figure 10: Auto-reply: ten candidate models on both benchmarks, under the pre-registered judge, ordered by real-benchmark score. *Left*: judged overall score. *Right*: critical-error rate. Ten thread prefixes per side; Phi-4 Reasoning has $n = 9$ on the real side, one generation call having failed.

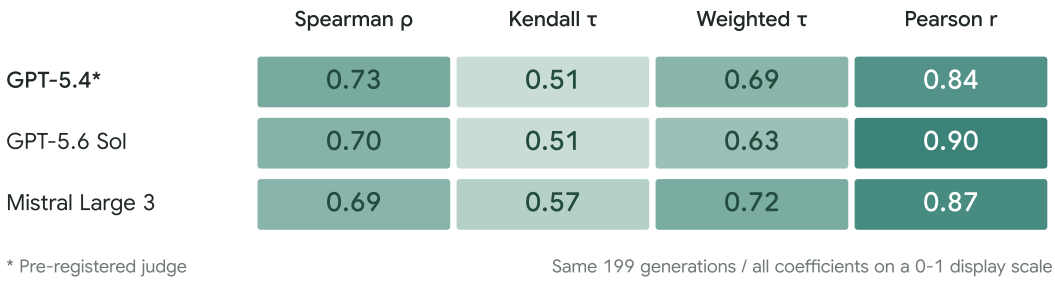


Figure 11: The same ten models, re-scored by three independent judges. Each judge rescores the identical stored generations, so the judge is the only thing that varies; all three reproduce the rank agreement between the two benchmarks. $*$ marks the pre-registered judge.