LOCAL-FIRST INTELLIGENCE VIA PROACTIVE SLM COLLABORATING WITH FEEDBACK-PROVIDING LLM

Anonymous authors
Paper under double-blind review

ABSTRACT

Personal intelligence requires strong capabilities under strict privacy and efficiency constraints. Unlike cloud-first intelligence, which relies on powerful but costly and privacy-vulnerable large language models (LLMs), we focus on local-first intelligence, emphasizing on-device small language models (SLMs), which enable efficient, private inference but have limited capacity. To reconcile the capability gap with privacy and efficiency, we propose a dynamic collaboration framework, where an SLM learns to proactively decide how to consult an LLM during multi-step reasoning, while the LLM provides adaptive feedback instead of acting as a passive tool, modeling them as heterogeneous agents under information asymmetry. We further systematically investigate how collaboration strategies are shaped by SLM and LLM capabilities as well as efficiency and privacy constraints. Evaluation results reveal a distinct scaling effect: stronger SLMs become more self-reliant, while stronger LLMs enable fewer and more informative interactions. In addition, the learned dynamic collaboration strategies significantly outperform various baselines and transfer robustly to unseen LLMs as well as to real-world personal tasks without additional training.

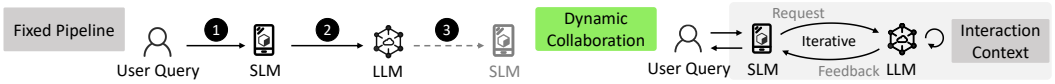
1 INTRODUCTION

Personal intelligence is emerging as a key paradigm for delivering context-aware assistance grounded in user-specific data, requiring strong performance, robust privacy preservation, and efficient interaction, as reflected in commercial systems like Gemini Personal Intelligence (Google, 2026) and Apple Intelligence (Apple Inc., 2026). Existing cloud-first intelligence prioritizes powerful large language models (LLMs), achieving strong performance but introducing significant challenges in privacy and cost. This motivates local-first intelligence, where on-device small language models (SLMs) serve as the first point of contact for user queries, offering lower cost and stronger privacy guarantees. However, local SLMs generally lack the strong capability and broad knowledge required to handle complex queries.

To enable powerful local-first intelligence, a natural solution is to integrate on-device SLMs with cloud-based LLMs. Existing approaches typically rely on static interactions (Yue et al., 2024; Zhang et al., 2024a; Ngong et al., 2025), which can be categorized by how user queries are exposed to the LLM. The first category directly sends unresolved queries or sub-queries to the LLM, as in AutoMix (Aggarwal et al., 2024) and Router-R1 (Zhang et al., 2025). However, these methods either ignore efficiency or rely on predefined heuristics, leading to unnecessary or missed LLM calls while overlooking privacy concerns. The second category masks private information before forwarding queries to the LLM, as in Self-Disclosures (Dou et al., 2024), yet static privacy filtering may either remove critical information and degrade answer quality, or fail to fully filter sensitive content. The third category rewrites queries with SLM before invoking the LLM and post-processes its responses, as in PAPILLON (Li et al., 2025). However, fixed interaction policies and limited awareness of the LLM's capabilities prevent the SLM from adaptively deciding when and what to consult, causing irrelevant or insufficient LLM calls. Overall, these methods treat the LLM as an external passive tool invoked through predefined pipelines, precluding the bidirectional interaction where the SLM adaptively requests assistance and the LLM provides targeted knowledge or feedback.

In this work, we propose a novel dynamic collaboration framework where both on-device SLMs and cloud-based LLMs are modeled as inherently heterogeneous agents with different capabilities and

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.



(a) Static Interaction Framework (b) Dynamic Collaboration Framework

Figure 1: Static interaction framework in existing work vs. Our dynamic collaboration framework.

asymmetric information. As illustrated in Figure 1, the SLM serves as the lead coordinator, first receiving the user query, conducting iterative reasoning, and adaptively deciding whether to proceed locally or consult the LLM. Ideally, the SLM formulates targeted requests only when assistance is needed and completes the reasoning independently otherwise. Crucially, the LLM is not treated as a black-box tool but as a flexible source of feedback and knowledge that participates in a multi-turn interaction with the SLM. At each interaction step, the LLM can take one of two actions: (1) provide feedback when the request is ambiguous, or (2) return relevant and useful information. We train the SLM through end-to-end online reinforcement learning (RL), enabling it to autonomously learn interaction timing and request formulation, conditioned on task difficulty and LLM feedback.

To understand the factors shaping the collaboration strategies, we systematically investigate different research questions. We evaluate how the SLM and LLM capabilities affect the effectiveness and dynamics of collaboration. We also examine whether the learned strategies enable richer and more effective interaction patterns than static interaction frameworks, whether they transfer to real-world personal tasks, whether SLMs can acquire privacy-preserving collaboration behaviors without sacrificing performance, and how external efficiency and privacy constraints affect strategies.

Our findings reveal several key insights. SLM and LLM capabilities shape collaboration strategies: stronger SLMs become more selective and self-reliant, while more capable LLMs support more concise and effective interactions. Notably, the learned strategies transfer effectively to unseen LLMs and further generalize to real-world personal tasks. Moreover, compared to static interaction frameworks, dynamic collaboration enables more efficient behavior by selectively invoking the LLM only when necessary, leading to improved performance and stronger privacy preservation.

We summarize our key contributions as follows: (1) We formulate local-first personal intelligence as a novel dynamic collaboration problem between on-device SLMs and cloud-based LLMs with heterogeneous capabilities and asymmetric information. (2) We propose a proactive SLM that adaptively decides when and how to collaborate, a collaborative LLM with a closed-loop feedback mechanism, and a multi-objective reward jointly optimizing response quality, efficiency, and privacy. (3) We construct the PrivQA, a benchmark of 153K privacy-augmented QA spanning general and multi-hop questions, and conduct systematic experiments, providing actionable insights into collaboration behaviors. Our method consistently improves EM by 3.0% – 15.6%, reduces interaction turns by 0.11 – 0.27, and lowers privacy leakage by 20.9% – 74.1% over static baselines. The learned policy also directly transfers to real-world personal tasks without additional training, improving quality by 3.1% – 6.9% and lowering privacy leakage by 5.4% – 26.0% over baselines.

2 Related Work

Collaboration between SLM and LLM has been studied mainly under two paradigms: cloud-first and local-first intelligence. Cloud-first approaches rely on LLMs as the primary inference engine, augmenting them with tools or domain-specific small models (Jin et al., 2025) or routing among multiple LLMs based on query characteristics (Zhu et al., 2023; Ding et al., 2024; Chen et al., 2024a; Ong et al., 2025). While effective, these methods remain centered on cloud-based LLMs and therefore inherit their privacy risks and efficiency concerns. Local-first approaches instead place an SLM on the user side as the primary interface. Most existing methods adopt a fixed pipeline in which the SLM preprocesses the query once and forwards it to the LLM (Li et al., 2020; Xia et al., 2023; Zhang et al., 2024a;b), either through query rewriting, as in PAPILLON (Li et al., 2025) and PrivacyRestore (Ngong et al., 2025), or through privacy filtering (Zhang et al., 2024c; Zeng et al., 2025), exemplified by Self-Disclosures (Dou et al., 2024). A more recent line of work trains the SLM as an agentic router: Privacy-R1 (Hui et al., 2026) dispatches non-sensitive chunks to the LLM, and Router-R1 (Zhang et al., 2025) routes sub-queries across multiple LLMs. Despite differing implementations, existing methods rely on a fixed interaction framework, treat the LLM as a passive tool, and address at most two of the three coupled requirements: response quality, interaction efficiency, and privacy preservation. In contrast, we propose a dynamic framework with closed-loop feedback

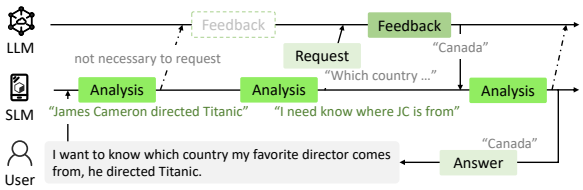


Figure 2: Illustration of dynamic collaboration workflow between proactive SLM and cloud-based LLM.

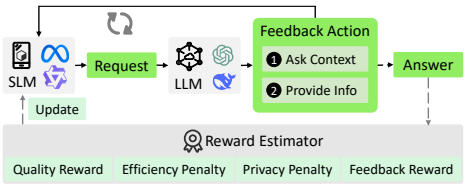


Figure 3: Training with LLM feedback via outcome- and process-based rewards.

in which the proactive SLM adaptively decides when and how to consult, and the LLM can either provide information or ask for context, jointly optimizing all three requirements under information asymmetry between heterogeneous agents. Appendix B.2 elaborates these methodological deltas.

Another related line of work is Multi-Agent Systems (MASs), where autonomous entities interact in a shared environment to achieve individual or collective goals (Dafoe et al., 2020; Soorati et al., 2022; Li et al., 2023; Tran et al., 2025). LLM-driven MASs have enabled agents to reason, plan, and communicate in natural language (Aher et al., 2023; Guo et al., 2024; Huang et al., 2024), as exemplified by MetaGPT (Hong et al., 2024) and AgentVerse (Chen et al., 2024b). These systems typically require agents to share task descriptions, execution traces, or environmental information. For local-first collaboration, however, models are heterogeneous in capability and subject to information asymmetry, as the on-device SLM and cloud-based LLM differ in capability, efficiency, and privacy exposure, and the LLM only accesses information selectively disclosed by the SLM. Appendix B further provide a broader comparison with existing work.

3 Problem Formulation

3.1 Workflow Formulation

We focus on local-first intelligence, a paradigm in which on-device SLMs act as the primary interface for user queries. As shown in Figure 2, when the SLM receives a query, it enters an iterative multi-step reasoning loop: at each step, it assesses whether the current subproblem can be solved locally with high confidence, and otherwise requests the required information from a more capable cloud-based LLM. After the LLM responds and provides feedback, the SLM integrates the returned information and continues solving subsequent steps until it produces a final answer for the user. For example, given the privacy-sensitive query "I want to know which country my favorite director comes from, he directed Titanic," the SLM first identifies the subproblem "Who directed Titanic?" and answers "James Cameron" locally, then forwards the next subproblem "Which country is James Cameron from?" to the LLM. After receiving "Canada," the SLM integrates the information and returns the final response. Throughout this process, only the SLM has access to the user's query and decides how to interact with the LLM.

We model the workflow in which an SLM autonomously requests assistance from an LLM as a collaborative process between heterogeneous agents with asymmetric capabilities and information availability. Formally, let M_s denote the on-device SLM and M_l denote the cloud-based LLM. Given a user query Q, which is available only to M_s, the SLM's behavior is modeled as a sequence of triplets $\langle \tau, a, o \rangle$. At step t, τ_t is the reasoning process, a_t denotes an action available to M_s, including making structured requests as well as generating the answer, and o_t is the observation received after each action, which consists of feedback provided by M_l. The trajectory terminates when M_s produces the final answer. For M_s, the trajectory history at step t is defined as

$$\mathcal{H}_t = \{\tau_0, a_0, o_0, \tau_1, a_1, o_1, \ldots, \tau_{t-1}, a_{t-1}, o_{t-1}\}.$$

M_s generates τ_t and a_t according to the collaboration policy given $\mathcal{H}_t$ and Q, and M_l generates o_t based on a_t issued by M_s and the interaction context between M_s and M_l.

3.2 Overview of Collaboration Goal

Local-first intelligence naturally poses three coupled objectives for the autonomous collaboration between SLMs and LLMs, namely delivering high-quality responses, interacting efficiently, and

minimizing privacy leakage. Response quality is evaluated by comparing the SLM's predictions with ground-truth answers for each user query. Interaction efficiency is analyzed in terms of the average number of interaction turns and the necessity of the SLM's requests to the LLM. Privacy leakage is measured by assessing whether any requests issued by the SLM expose information from the set of user private data. Beyond these optimization objectives, we further investigate how different SLMs autonomously learn effective collaboration strategies with different LLMs under tasks with varying difficulty and constraints. We systematically examine how various factors influence the collaboration strategies adopted by SLMs.

4 PROACTIVE SLM WITH LLM FEEDBACK

We design dynamic collaboration between heterogeneous SLM and LLM under information asymmetry, where SLM learns to proactively consult LLM with active feedback, as shown in Figure 3.

4.1 OUTCOME QUALITY AND INTERACTION NECESSITY DRIVEN SLM REQUEST

The SLM is trained to learn a collaboration policy $\pi_\theta(y \mid x, M_l)$, where $x = Q$ is the user query and $y = (\tau_t, a_t, o_t)_{t=0}^{T}$ is the interaction trajectory, with τ_t and a_t generated by M_s and o_t returned by M_l. It must decide when to query the LLM and how to formulate each request, based on the respective capabilities of M_s and M_l and the information required for the task, jointly optimizing response quality and interaction efficiency.

For response quality, we adopt an outcome-based quality reward r_{quality} to encourage the SLM to produce high-quality answers. Formally, r_{quality} is defined as the Exact Match (EM) score between the prediction ans_{pred} and the ground-truth ans_{gt} as follows.

$$r_{\text{quality}} = \text{EM}(ans_{\text{pred}}, ans_{\text{gt}})$$

For efficiency, the SLM is encouraged to issue requests only when necessary, i.e., when it genuinely lacks the required knowledge. We design a difficulty-aware efficiency penalty $r_{\text{efficiency}}$. **For easy queries, the SLM is expected to answer without issuing any requests. For hard queries, the SLM may issue requests but should minimize the number of interaction turns to obtain the correct answer.** We employ a simple but effective difficulty judgment strategy: if the SLM can correctly answer the user query without relying on external information, the query is considered easy; otherwise, it is considered hard. Formally, $r_{\text{efficiency}}$ is defined as

$$r_{\text{efficiency}} = -\mathbb{I}(Q \notin D_{\text{e}})\frac{n_r}{n_{max}} - \mathbb{I}(Q \in D_{\text{e}} \wedge n_r > 0),$$

where $Q \in D_{\text{e}}$ indicates that the query is easy for the current SLM, n_r denotes the total number of interaction turns, and $n_{\max}$ denotes the predefined maximum number of interaction turns.

4.2 ACTIVE LLM FEEDBACK

Prior collaboration frameworks treat the LLM as a passive tool that returns information for whatever request the SLM sends. This is problematic when the SLM, constrained by its limited capability, issues requests that are vague or under-specified: the LLM may return superficially relevant but ultimately noisy information, causing errors to accumulate over multiple turns. We instead model the LLM as an active collaborator whose response adapts to the precision of the SLM's request. At each interaction step t, the LLM takes the SLM's request a_t together with the interaction history and selects one of two actions: **(1) Clarify if a_t is under-specified, asking the SLM to provide the missing context; or (2) Answer if a_t is well-specified, returning the relevant information**. The selected action yields the observation o_t that is fed back into the SLM's trajectory. The action decision is made via a prompted rubric detailed in Appendix J and the LLM is not further trained.

Furthermore, the LLM's feedback also serves as a process-level learning signal for the SLM: **a request that prompts a clarification is by construction under-informative, so we penalize it through a feedback reward r_{feedback}.** For example, the request "I need to know where the user's cousin lives" would elicit a clarification and receive $r_{\text{feedback}} = -1$, teaching the SLM to issue more

concrete requests in subsequent turns. Formally, r_{feedback} is defined as

$$r_{\text{feedback}} = \begin{cases} -1, \text{ LLM asks contextual information} \\ 0, \text{ otherwise} \end{cases}$$

4.3 Privacy Awareness with Dataset Synthesis

To ensure privacy-preserving collaboration, the SLM needs to learn to formulate requests without disclosing sensitive user information. We address this by introducing a privacy penalty, r_{privacy}, which is applied to the interaction whenever an SLM request results in a privacy leak:

$$r_{\text{privacy}} = \begin{cases} -1, \text{ if the request leaks privacy} \\ 0, \text{ otherwise.} \end{cases}$$

Whether privacy is leaked is judged by a separate evaluator LLM (Zheng et al., 2023). Given the labeled private information in the user query and the requests generated by the SLM across all interaction turns, the evaluator LLM determines whether any private information can be inferred, ensuring multi-turn, compositional, and indirect leakage is covered. Under an honest-but-curious LLM adversary, Appendix G provides the full formulation. Human evaluation on 200 instances agrees with the evaluator's judgments 98.5% of the time. Appendix F details the protocol and further shows that judgments are consistent across evaluators from different model families.

To train the SLM's privacy awareness, it requires data containing user privacy information. However, existing privacy QA datasets (Jin et al., 2020; Li et al., 2025) contain only a few hundred samples primarily designed for evaluation and are limited in diversity. We therefore build a privacy injection pipeline that **augments existing QA data by prompting a synthesis LLM to replace explicit entities with privacy-related indirect references, simulating inadvertent user disclosure**. As shown in Figure 9, given an original QA pair, we first generate logically consistent personal information related to the core entities; we then reformulate the underlying factual statement so that the answer term is replaced by an indirect reference through this personal information; finally, we combine the reformulated fact with a new question derived from the personal information to obtain a privacy-infused QA pair with an unchanged correct answer. The injected information is thus a necessary premise for reaching the answer, not a separable decoration. The synthesis LLM subsequently filters cases where the answer is altered or explicitly revealed in the question. We further conduct human evaluation on 200 randomly sampled instances, confirming that 99% are correctly reformulated with accurate answers and appropriate privacy infusion.

Applying this pipeline, we build the Privacy QA (PrivQA) dataset from six public QA sources: three general-domain datasets (NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2022)) and three multi-hop datasets (HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), MuSiQuE (Trivedi et al., 2022)), which are collected from genuine user queries such as Google search logs and TREC. Each PrivQA instance contains the privacy-augmented question, its ground-truth answer, and the synthesized private information used for quantitative leakage evaluation. Appendix C presents the pipeline details and dataset statistics.

4.4 Collaboration Strategy Learning

We finally integrate the goals of response quality, efficiency, and privacy into a multi-objective reward function. We additionally include a format reward $r_{\text{format}} \in \{-1, 0\}$ that penalizes format violations, as shown in Appendix E. The final reward function is defined as

$$R = r_{\text{format}} + \underbrace{r_{\text{quality}} + r_{\text{feedback}}}_{\text{answer reward}} + \underbrace{\alpha r_{\text{privacy}} + \beta r_{\text{efficiency}}}_{\text{constraint penalty}}.$$

where $\alpha, \beta \geq 0$ trade off quality against privacy and efficiency. Larger weights induce more privacy-preserving or more efficient collaboration, allowing the framework to adapt to different deployment requirements. Appendix H proves that reward judgment errors do not change the gradient direction.

Given the reward, the SLM is optimized using the latest algorithm, VAPO (Yue et al., 2025). The policy model is updated to maximize the following objective:

$$\mathcal{J}_{\text{VAPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_{\theta_{\text{old}}}(\cdot|x;o)}\left[\frac{1}{\sum_{t=1}^{|y|}\mathbb{I}(y_t \neq o)}\sum_{t=1}^{|y|}\min\left(p_t(\theta)\hat{A}_t, \text{clip}\Big(p_t(\theta), 1-\varepsilon_{low}, 1+\varepsilon_{high}\Big)\hat{A}_t\right)\right],$$

where $p_t(\theta) = \frac{\pi_\theta\left(y|x, y_{<t}; o\right)}{\pi_{\theta_{\text{old}}}\left(y|x, y_{<t}; o\right)}$ is the probability ratio, the terms ε_{low} and ε_{high} are the lower and higher clipping range hyperparameter, and $\hat{A}_t$ denotes the estimated advantage at time step t, calculated using Length-Adaptive GAE (Yue et al., 2025) based on future rewards $R \geq t$ and a learned value model. π_θ and $\pi_{\theta_{\text{old}}}$ are the current and previous policy models, respectively. The indicator function $\mathbb{I}(y_t \neq o)$ masks the token loss for o generated by M_l.

5 Evaluation

5.1 Experimental Setup

Model setup. For the SLMs, we employed Qwen3-4B and Qwen3-8B (Yang et al., 2025) with reasoning capabilities, as well as Llama-3.2-3B-Instruct and Llama-3.1-8B-Instruct (Grattafiori, A., et al., 2024). For the LLMs, we utilized the open-source DeepSeek-R1-Distill-Llama-70B (DeepSeek-AI, 2025) and Qwen3-235B-A22B-Instruct, along with the closed-source Qwen3-Max. We pair different SLMs and LLMs for systematic evaluation.

Dataset setup. We train the SLM on the PrivQA dataset, which consists of privacy-augmented QA samples sourced from NQ, TriviaQA, PopQA, HotpotQA, 2Wiki, and MuSiQuE. To thoroughly evaluate the generalization capability of the learned collaboration strategy, we construct the training set using only samples derived from NQ and HotpotQA, while the test set covers all six datasets. We additionally evaluate the learned policy on three real-world benchmarks without any further training, namely PUPA (Li et al., 2025) for email and document writing, AESLC (Zhang & Tetreault, 2019) for email subject-line writing, and SAMSum (Gliwa et al., 2019) for dialogue summarization.

Evaluation metrics. For quality, we employ Exact Match (EM) and BERTScore (F_{BERT}) (Zhang et al., 2020) on PrivQA. On the three real-world personal tasks, which are open-ended generation tasks, we adopt LLM-as-a-judge with prompted rubrics (Zheng et al., 2023).

For efficiency, we report the average number of interaction turns. We also introduce the Interaction Necessity Score (INScore), a difficulty-aware metric. Queries are labeled easy or hard depending on whether the SLM can answer them independently. Using difficulty as ground truth and the SLM's request decision as prediction, INScore is the average of two recalls: the proportion of easy queries handled locally and the proportion of hard queries for which assistance is requested. Since easy or hard labels are calibrated per SLM, INScore is a within-SLM efficiency metric.

For privacy, we compute the average semantic similarity between the SLM's requests and the set of privacy information with BERTScore (Priv_{sim}) (Ngong et al., 2025) and measure the proportion of samples where the SLM issues privacy-leaking requests ($\text{Priv}_{\text{sample}}$) with an LLM judge (Li et al., 2025). Detailed definitions of metrics are provided in Appendix G.

Baselines. We compare against three groups. **Non-collaboration.** (1) SLM with Chain of Thought (CoT) (Wei et al., 2022); (2) direct LLM inference (Brown et al., 2020); (3) LLM with CoT, the LLM-side upper bound; (4) SFT on PrivQA data, answering locally. **Static collaboration.** (5) AutoMix (Aggarwal et al., 2024), forwarding queries the SLM cannot resolve; (6) Self-Disclosures (Dou et al., 2024), masking private tokens before forwarding; (7) PAPILLON (Li et al., 2025), SLM query rewriting followed by an LLM answer and SLM aggregation; (8) Router-R1 (Zhang et al., 2025), an SLM trained via RL to dispatch sub-queries. **Ablation.** (9) Ours w/o Feedback, our framework in which the LLM acts as a passive answer-only tool, trained with the same reward.

Implementation details. We implement all designs with verl framework (Sheng et al., 2025). The workstation has 8 H20 96G GPUs. More details are listed in Appendix I.1.

5.2 Gains of Dynamic Collaboration

RQ1. Does the learned collaboration strategy lead to better interaction patterns than baselines across both PrivQA and unseen real-world personal tasks?

Our dynamic collaboration framework simultaneously improves response quality, privacy preservation, and interaction efficiency over all baselines. Table 1 reports results on PrivQA. Static baselines lag standalone LLM inference, e.g., Router-R1 with Qwen3-8B is 4.5% lower in

Table 1: Quality, privacy leakage, and efficiency performance on the PrivQA dataset, including in- and out-domain datasets. **Bold** denotes the best result and underline the second-best.

Method	Model	General QA			Multi-Hop QA			Avg.		
		EM ↑	$\text{Priv}_{\text{sample}}$ ↓	INScore ↑	EM ↑	$\text{Priv}_{\text{sample}}$ ↓	INScore ↑	EM ↑	$\text{Priv}_{\text{sample}}$ ↓	INScore ↑
CoT	Qwen3-235B	50.89	100.00	50.00	34.11	100.00	50.00	42.50	100.00	50.00
SFT	Qwen3-4B	25.73	0.00	50.00	19.70	0.00	50.00	22.72 $_{(-15.6)}$	0.00 $_{(+0.3)}$	50.00 $_{(-9.6)}$
	Qwen3-8B	33.74	0.00	50.00	21.85	0.00	50.00	27.80 $_{(-15.3)}$	0.00 $_{(+0.1)}$	50.00 $_{(-10.9)}$
AutoMix	Qwen3-4B	37.65	72.34	58.44	25.93	76.28	54.78	31.79 $_{(-6.6)}$	74.31 $_{(-74.1)}$	56.61 $_{(-3.0)}$
	Qwen3-8B	41.10	66.12	59.51	26.45	71.68	56.62	33.78 $_{(-9.3)}$	68.90 $_{(-68.8)}$	58.06 $_{(-2.8)}$
Self-Disclosures	Qwen3-4B	32.57	59.60	50.00	18.43	57.59	50.00	25.50 $_{(-12.8)}$	58.60 $_{(-58.3)}$	50.00 $_{(-9.6)}$
	Qwen3-8B	36.08	53.86	50.00	22.09	49.22	50.00	29.09 $_{(-14.0)}$	51.54 $_{(-51.5)}$	50.00 $_{(-10.9)}$
PAPILLON	Qwen3-4B	41.43	34.99	50.00	23.68	30.30	50.00	32.55 $_{(-5.8)}$	32.64 $_{(-32.4)}$	50.00 $_{(-9.6)}$
	Qwen3-8B	42.55	23.88	50.00	23.91	24.87	50.00	33.23 $_{(-9.9)}$	24.37 $_{(-24.3)}$	50.00 $_{(-10.9)}$
Router-R1	Qwen3-4B	43.68	30.25	52.32	27.06	27.34	53.28	35.37 $_{(-3.0)}$	28.80 $_{(-28.5)}$	52.80 $_{(-6.8)}$
	Qwen3-8B	47.29	22.04	55.78	<u>28.62</u>	19.99	55.27	37.96 $_{(-5.2)}$	21.01 $_{(-20.9)}$	55.52 $_{(-5.4)}$
Ours	Qwen3-4B	<u>48.47</u>	<u>0.18</u>	<u>60.98</u>	28.21	<u>0.33</u>	<u>58.17</u>	<u>38.34</u> $_{(+0.0)}$	<u>0.26</u> $_{(+0.0)}$	<u>59.57</u> $_{(+0.0)}$
	Qwen3-8B	**52.86**	**0.08**	**61.82**	**33.35**	**0.08**	**59.99**	**43.11** $_{(+0.0)}$	**0.08** $_{(+0.0)}$	**60.91** $_{(+0.0)}$

EM. In contrast, our framework matches or exceeds it. Compared to baselines, on Qwen3-8B, our method improves EM by 5.2% – 15.3%. In terms of privacy, our method almost eliminates leakage: $\text{Priv}_{\text{sample}}$ is only 0.26% on Qwen3-4B and 0.08% on Qwen3-8B, far below the best baselines of 28.80% and 21.01%, respectively. This is attributed to the SLM's ability to reformulate private queries into objective, user-agnostic requests. Moreover, by adaptively deciding when to request assistance, our method improves INScore on Qwen3-8B over the baselines by 2.8% – 10.9%.

Our framework effectively transfers to real-world personal tasks without additional training. Table 2 reports the results on the three real-world benchmarks, PUPA, AESLC, and SAMSum. With Qwen3-8B, our framework achieves the lowest average privacy leakage rate 1.01% and the highest average quality score 92.11% among all methods. Specifically, our method reduces average leakage by 6.19% – 26.01% on Qwen3-4B and 5.35% – 21.16% on Qwen3-8B, and improves quality by 3.07% – 6.17% and 4.00% – 6.87%, respectively, over baselines. This confirms that the collaboration policy learned on PrivQA generalizes well to unseen real-world datasets and user scenarios.

Table 2: Average performance and privacy leakage on real-world personal tasks.

Method	SLM	Avg.	
		Quality. ↑	$\text{Priv}_{\text{sample}}$ ↓
CoT	Qwen3-235B	95.21	100.00
AutoMix	Qwen3-4B	82.07 $_{(-4.9)}$	27.11 $_{(-26.0)}$
	Qwen3-8B	86.96 $_{(-5.2)}$	22.17 $_{(-21.2)}$
Self-Disclosures	Qwen3-4B	80.77 $_{(-6.2)}$	23.20 $_{(-22.1)}$
	Qwen3-8B	85.24 $_{(-6.9)}$	18.98 $_{(-18.0)}$
PAPILLON	Qwen3-4B	83.87 $_{(-3.1)}$	8.55 $_{(-7.5)}$
	Qwen3-8B	<u>88.11</u> $_{(-4.0)}$	6.68 $_{(-5.7)}$
Router-R1	Qwen3-4B	82.76 $_{(-4.2)}$	7.29 $_{(-6.2)}$
	Qwen3-8B	88.00 $_{(-4.1)}$	6.36 $_{(-5.4)}$
Ours	Qwen3-4B	86.94 $_{(+0.0)}$	<u>1.10</u> $_{(+0.0)}$
	Qwen3-8B	**92.11** $_{(+0.0)}$	**1.01** $_{(+0.0)}$

5.3 Impact of Different SLMs

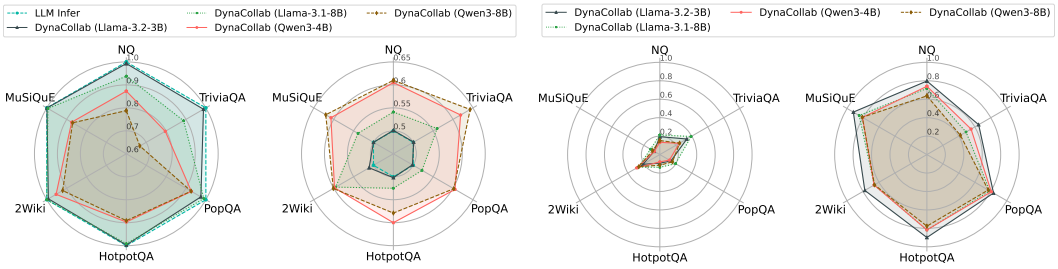
RQ2. How does the capability of the SLM affect the collaboration strategies?

Stronger SLM achieves better collaboration performance, while smaller SLM exhibits greater improvement. Table 3 reports the performance of different SLMs under dynamic collaboration. Across all datasets, our framework outperforms standalone SLM inference, and stronger SLMs perform better. In particular, dynamic collaboration improves EM over direct LLM inference by 1.4% for Qwen3-4B and 6.1% for Qwen3-8B, while Qwen3-8B further exceeds LLM with CoT by 0.6% on average EM. These results confirm that SLMs learn effective coordination with the LLM. Within the same family, smaller SLMs gain more from collaboration: Llama-3.2-3B improves 18.1% EM over its own CoT versus 14.5% for Llama-3.1-8B, narrowing the gap across SLM scales.

Table 3: Average performance of different SLMs with Qwen3-235B and baselines.

Method	SLM	Avg.	
		EM ↑	F_{BERT} ↑
Direct Infer	Qwen3-235B	36.96	70.54
CoT	Qwen3-235B	42.50	74.54
CoT	Llama-3.2-3B	17.74	56.49
	Llama-3.1-8B	24.38	60.86
	Qwen3-4B	20.96	62.58
	Qwen3-8B	25.77	64.73
Ours	Llama-3.2-3B	35.86$_{(+18.1)}$	71.42$_{(+14.9)}$
	Llama-3.1-8B	<u>38.92</u>$_{(+14.5)}$	73.37$_{(+12.5)}$
	Qwen3-4B	38.34$_{(+17.4)}$	<u>73.51</u>$_{(+10.9)}$
	Qwen3-8B	**43.11**$_{(+17.3)}$	**75.55**$_{(+10.8)}$

SLMs with fewer parameters rely more heavily on LLM to improve their final performance. As shown in Figure 4, compared with directly forwarding user queries to the LLM, our framework reduces average interaction turns by 0.11 and 0.15, while improving INScore by 9.6% and 10.9% on Qwen3-4B and Qwen3-8B, respectively. Across model scales, smaller SLMs rely more on the LLM: Llama-3.2-3B and Qwen3-4B increase average interaction turns by 0.14 and 0.04 over Qwen3-

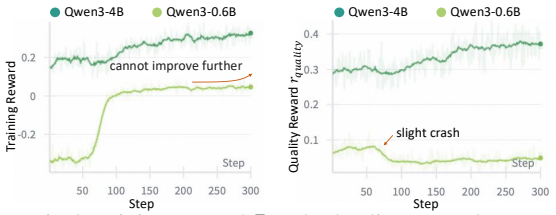


(a) Avg Interaction Turn ↓ (b) INScore ↑ (c) Request Rate (Easy) (d) Request Rate (Hard)

Figure 4: Efficiency analysis of collaboration strategy learned by different SLMs with same LLM.

8B, and Llama-based SLMs request LLM assistance more often than Qwen-based counterparts. Moreover, request rates are much lower on easy samples than on hard ones. These findings show that SLMs adaptively decide when to request based on task difficulty and their own capabilities.

SLMs with weak instruction-following capabilities fail to receive effective feedback under end-to-end RL. To assess whether all SLMs can learn collaboration via end-to-end RL, we further evaluate Qwen3-0.6B. In Figures 5a and 5b, Qwen3-0.6B converges to a low final reward with a mild collapse in r_{quality}, while Qwen3-4B achieves a stable, monotonic reward increase. This gap stems from Qwen3-0.6B's weak instruction-following ability: it rarely produces well-formed LLM requests during exploration, eliciting unhelpful responses and limited positive rewards. Such SLMs therefore require a supervised cold-start to teach basic request formatting before RL.



(a) Final Training Reward R (b) Quality Reward r_{quality}

Figure 5: Training reward and quality reward of 0.6B and 4B models collaborating with Qwen3-235B.

5.4 Impact of Different LLMs

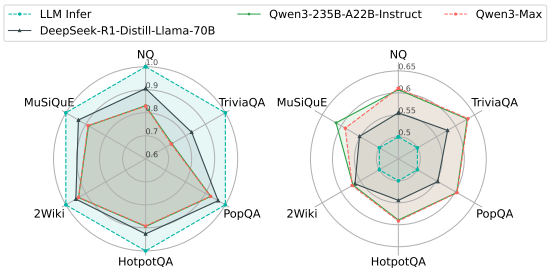
RQ3. How does LLM capability affect the effectiveness and behavior of collaboration?

Collaborating with stronger LLMs yields better performance. Table 4 reports the performance of identical SLMs paired with different LLMs. Across different LLMs, SLMs learn effective collaboration strategies and achieve performance comparable to their LLM partners. Moreover, we observe that stronger LLMs provide more informative and relevant interaction signals, which lead to more accurate answers. For Qwen3-4B and Qwen3-8B, collaborating with Qwen3-235B yields average EM improvements of 1.8% and 5.3%, respectively, over DeepSeek-Distill-Llama-70B.

Table 4: Average performance of SLM with different LLMs.

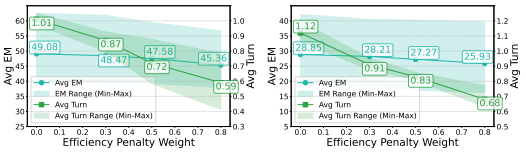
Method	SLM	LLM	Avg.	
			EM ↑	F_{BERT} ↑
CoT	no SLM	DeepSeek-70B	39.95	72.94
		Qwen3-235B	42.50	74.54
		Qwen3-Max	47.08	76.89
Ours	Qwen3-4B	DeepSeek-70B	36.57	72.56
	Qwen3-8B		37.80	73.31
	Qwen3-4B	Qwen3-235B	38.34	73.51
	Qwen3-8B		<u>43.11</u>	<u>75.55</u>
	Qwen3-4B	Qwen3-Max*	41.68	75.13
	Qwen3-8B		**46.01**	**77.02**

Fewer interaction turns are needed when collaborating with stronger LLMs. To analyze how SLMs adapt their collaboration strategies to different LLMs, we compare the average interaction turns and INScore across different LLM partners in Figure 6a and Figure 6b. For the same SLM, collaboration with a stronger LLM generally requires fewer interaction turns. For example, pairing Qwen3-4B with Qwen3-235B reduces the average turns by 0.10 compared with DeepSeek-Distill-Llama-70B. The key observation is that when paired with the weaker



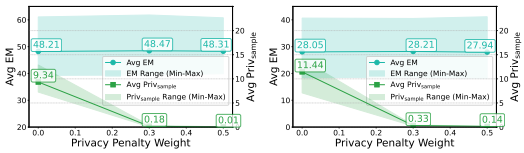
(a) Interaction Turn ↓ (b) INScore ↑

Figure 6: Average interaction turn and interaction necessity score across SLMs with different LLMs.



(a) General QA (b) Multi-hop QA

Figure 7: Pareto frontier of interaction efficiency and EM under varying efficiency weight.



(a) General QA (b) Multi-hop QA

Figure 8: Pareto frontier of $\text{Priv}_{\text{sample}}$ and EM under varying privacy weight

LLM, the SLM tends to decompose the problem into multiple sub-requests, whereas with a stronger LLM, it often issues a single consolidated request.

Collaboration strategies learned with a smaller LLM transfer effectively to larger unseen LLMs. SLMs trained via collaboration with Qwen3-235B-A22B-Instruct directly generalize to Qwen3-Max, an unseen closed-source LLM, and yield further improvements. Specifically, using Qwen3-Max improves average EM by 3.3% for Qwen3-4B and 2.9% for Qwen3-8B over Qwen3-235B-A22B-Instruct, while the average number of interaction turns remains largely unchanged. This suggests that we can train collaboration behaviors with a smaller LLM and deploy with a stronger one at inference time, reducing training cost while further improving performance.

5.5 Ablation Study and Hyper-Parameter Analysis

RQ4. How do the external constraints shape the learned collaboration strategies?

Impact of LLM Feedback. We isolate the contribution of active LLM feedback by an ablation that replaces the feedback LLM with a passive answer-only tool while keeping the reward identical. As shown in Table 5, removing feedback drops average EM by 2.8% and F_{BERT} by 2.0% on Qwen3-4B. This gap indicates that the clarify action supplies a process-level supervision signal on request quality that outcome reward alone cannot substitute, localizing errors to individual under-specified requests rather than attributing them to an entire multi-turn trajectory. Per-dataset results are in Appendix Table 20. Appendix I.9 further reports the frequency of the LLM's two feedback actions.

Table 5: Average performance of Qwen3-4B interacting with a feedback LLM vs. a tool LLM.

Method	Model	Avg.	
		EM ↑	F_{BERT} ↑
CoT	Qwen3-235B	42.50	74.54
Ours w/o feedback	Qwen3-4B	35.53	71.52
Ours	Qwen3-4B	**38.34**	**73.51**

Impact of Efficiency Penalty. Figure 7 shows how the efficiency penalty weight controls the trade-off between interaction and quality. As the weight increases, the SLM relies more on local reasoning and requests the LLM less frequently. For instance, setting the weight to 0.8 reduces the average interaction turns between Qwen3-4B and Qwen3-235B by 0.43 at the cost of a 3.3% drop in EM. This clear trade-off between efficiency and answer quality can be tuned via the efficiency penalty at training time, allowing the SLM to adapt to different cost-sensitive deployment scenarios.

Impact of Privacy Penalty. Figure 8 shows the effect of privacy penalties on collaboration strategies. Without privacy constraints, the SLM partially leaks user information because it prioritizes information retrieval over privacy reformulation. Setting the privacy penalty weight above 0.3 produces strong privacy protection: relative to no penalty, $\text{Priv}_{\text{sample}}$ decreases by 10.1% with only a negligible drop in EM. This gain arises because the SLM learns to reformulate sensitive queries into objective, user-agnostic requests, preserving privacy while still retrieving the necessary information.

6 Conclusion

In this work, we have presented a dynamic collaboration framework for local-first personal intelligence, which transforms the interaction between on-device SLMs and cloud-based LLMs from static, predefined pipelines into a proactive, adaptive partnership. We have demonstrated that SLMs can autonomously acquire collaboration strategies that balance the comprehensive objectives of response quality, interaction efficiency, and privacy preservation, and that these strategies transfer to real-world writing and summarization tasks. By enabling heterogeneous models to communicate through informative feedback rather than simple query-response loops, we move closer to a future where local and cloud intelligence operate as a unified ecosystem.

AI USE STATEMENT

We used generative AI tools for two research subtasks: (i) constructing the PrivQA dataset (Section 4.4; Appendix C), where a synthesis LLM injects privacy-related indirect references into base QA pairs; and (ii) automated evaluation via LLM-as-a-judge (Zheng et al., 2023; Li et al., 2025), applied to privacy leakage on PrivQA and to response quality on the real-world benchmarks (Appendix G). The reliability of both uses is validated by human evaluation in Appendix F. Beyond these subtasks, generative AI tools were not used for formulating hypotheses, developing theoretical models, designing experiments, or interpreting results. We take full responsibility for all content of this work.

ETHICS STATEMENT

Our work targets a privacy-preserving personal-intelligence setting and intersects with the Code of Ethics on privacy, dataset release, and human-subject studies, which we address as follows. First, PrivQA is constructed by injecting *fictional* privacy items into publicly released QA datasets; we introduce no real user identifiers, and the real-world benchmarks we evaluate on (Li et al., 2025; Zhang & Tetreault, 2019; Gliwa et al., 2019) are also publicly available. Second, the human evaluations reported in Appendix F involve only volunteer annotators from our lab reviewing model-generated text, and contain no personally identifying information. We have reviewed the Code of Ethics and adhered to it throughout the work.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we release the following alongside the paper. Our full training and inference stack is built on the verl framework (Sheng et al., 2025) using the VAPO algorithm (Yue et al., 2025); all reward weights, hyperparameters, and hardware are documented in Appendix I.1, with training and inference compute reported in Appendices D.2 and D.3. The PrivQA construction pipeline, dataset statistics, and synthesis cost are detailed in Appendix C and Appendix D.1, and every prompt used in training, dataset synthesis, and evaluation is provided verbatim in Appendix J. The convergence result under evaluator noise is proved in Appendix H, with all assumptions made explicit. All base models used in our experiments are publicly available. An anonymized source code is submitted as supplementary material.

REFERENCES

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. Automix: Automatically mixing language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Gati V. Aher, Rosa I. Arriaga, and Adam Tauman Kalai. Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 337–371. PMLR, 2023. URL https://proceedings.mlr.press/v202/aher23a.html.

Apple Inc. Apple intelligence. https://www.apple.com/sg/apple-intelligence/, 2026. Apple Intelligence and Siri.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Huaben Chen, Wenkang Ji, Lufeng Xu, and Shiyu Zhao. Multi-agent consensus seeking via large language models. *CoRR*, abs/2310.20151, 2023. doi: 10.48550/ARXIV.2310.20151. URL https://doi.org/10.48550/arXiv.2310.20151.

Shuhao Chen, Weisen Jiang, Baijiong Lin, James T. Kwok, and Yu Zhang. Routerdc: Query-based router by dual contrastive learning for assembling large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024a.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024b. URL https://openreview.net/forum?id=EHg5GDnyq1.

Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R. McKee, Joel Z. Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative AI. *CoRR*, abs/2012.08630, 2020. URL https://arxiv.org/abs/2012.08630.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=02f3mUtqnM.

Yao Dou, Isadora Krsek, Tarek Naous, Anubha Kabra, Sauvik Das, Alan Ritter, and Wei Xu. Reducing privacy risks in online self-disclosures with language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 13732–13754. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.741. URL https://doi.org/10.18653/v1/2024.acl-long.741.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006*, pp. 265–284, 2006.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization, Hong Kong, China, November 4, 2019*, pp. 70–79. Association for Computational Linguistics, 2019.

Google. Gemini introduces personal intelligence. https://blog.google/innovation-and-ai/products/gemini-app/personal-intelligence/, January 2026. Blog post, Gemini introduces Personal Intelligence.

Grattafiori, A., et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pp. 8048–8057. ijcai.org, 2024. URL https://www.ijcai.org/proceedings/2024/890.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online),*

December 8-13, 2020, pp. 6609–6625. International Committee on Computational Linguistics, 2020. doi: 10.18653/V1/2020.COLING-MAIN.580. URL https://doi.org/10.18653/v1/2020.coling-main.580.

Haoyang Hong, Jiajun Yin, Yuan Wang, Jingnan Liu, Zhe Chen, Ailing Yu, Ji Li, Zhiling Ye, Hansong Xiao, Yefei Chen, Hualei Zhou, Yun Yue, Minghui Yang, Chunxiao Guo, Junwei Liu, Peng Wei, and Jinjie Gu. Multi-agent deep research: Training multi-agent systems with M-GRPO. *CoRR*, abs/2511.13288, 2025. doi: 10.48550/ARXIV.2511.13288. URL https://doi.org/10.48550/arXiv.2511.13288.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for A multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. Understanding the planning of LLM agents: A survey. *CoRR*, abs/2402.02716, 2024. doi: 10.48550/ARXIV.2402.02716. URL https://doi.org/10.48550/arXiv.2402.02716.

Zheng Hui, Yijiang River Dong, Sanhanat Sivapiromrat, Ehsan Shareghi, and Nigel Collier. Privacy-r1: Privacy-aware multi-llm agent collaboration via reinforcement learning, 2026. URL https://arxiv.org/abs/2510.16054.

Emmi Ignatius and Marja Kokkonen. Factors contributing to verbal self-disclosure. *Nordic Psychology*, 59(4):362–391, 2007.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *CoRR*, abs/2503.09516, 2025. doi: 10.48550/ARXIV.2503.09516. URL https://doi.org/10.48550/arXiv.2503.09516.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *CoRR*, abs/2009.13081, 2020. URL https://arxiv.org/abs/2009.13081.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1601–1611. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-1147. URL https://doi.org/10.18653/v1/P17-1147.

Daniel Kifer and Ashwin Machanavajjhala. Pufferfish: A framework for mathematical privacy definitions. *ACM Trans. Database Syst.*, 39(1):3:1–3:36, 2014. doi: 10.1145/2514689. URL https://doi.org/10.1145/2514689.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466, 2019. doi: 10.1162/TACL_A_00276. URL https://doi.org/10.1162/tacl_a_00276.

Huao Li, Yu Quan Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Charles Lewis, and Katia P. Sycara. Theory of mind for multi-agent collaboration via large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 180–192. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.13. URL https://doi.org/10.18653/v1/2023.emnlp-main.13.

Siyan Li, Vethavikashini Chithrra Raghuram, Omar Khattab, Julia Hirschberg, and Zhou Yu. PAPILLON: privacy preservation from internet-based and local language model ensembles. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pp. 3371–3390. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.NAACL-LONG.173. URL https://doi.org/10.18653/v1/2025.naacl-long.173.

Yuanqi Li, Arthi Padmanabhan, Pengzhan Zhao, Yufei Wang, Guoqing Harry Xu, and Ravi Netravali. Reducto: On-camera filtering for resource-efficient real-time video analytics. In *SIGCOMM '20: Proceedings of the 2020 Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication, Virtual Event, USA, August 10-14, 2020*, pp. 359–376. ACM, 2020. doi: 10.1145/3387514.3405874. URL https://doi.org/10.1145/3387514.3405874.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *CoRR*, abs/2212.10511, 2022. doi: 10.48550/ARXIV.2212.10511. URL https://doi.org/10.48550/arXiv.2212.10511.

Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pp. 286–299. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10610855. URL https://doi.org/10.1109/ICRA57147.2024.10610855.

Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. Can LLMs keep a secret? testing privacy implications of language models via contextual integrity theory. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Ivoline C. Ngong, Swanand Ravindra Kadhe, Hao Wang, Keerthiram Murugesan, Justin D. Weisz, Amit Dhurandhar, and Karthikeyan Natesan Ramamurthy. Protecting users from themselves: Safeguarding contextual privacy in interactions with conversational agents. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 26196–26220. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-acl.1343/.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pp. 1279–1297. ACM, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Mohammad Divband Soorati, Enrico H. Gerding, Enrico Marchioni, Pavel Naumov, Timothy J. Norman, Sarvapali D. Ramchurn, Bahar Rastegari, Adam J. Sobey, Sebastian Stein, Danesh Tarpore, Vahid Yazdanpanah, and Jie Zhang. From intelligent agents to trustworthy human-centred multiagent systems. *AI Commun.*, 35(4):443–457, 2022. doi: 10.3233/AIC-220127. URL https://doi.org/10.3233/AIC-220127.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. Multi-agent collaboration mechanisms: A survey of llms. *CoRR*, abs/2501.06322, 2025. doi: 10.48550/ARXIV.2501.06322. URL https://doi.org/10.48550/arXiv.2501.06322.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554, 2022. doi: 10.1162/TACL_A_00475. URL https://doi.org/10.1162/tacl_a_00475.

Jingkang Wang, Yang Liu, and Bo Li. Reinforcement learning with perturbed rewards. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 6202–6209. AAAI Press, 2020.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pp. 3909–3925. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.257. URL https://doi.org/10.18653/v1/2023.findings-emnlp.257.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1259. URL https://doi.org/10.18653/v1/d18-1259.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thought representations for cost-efficient reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=6okaSfANzh.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengxiang Wang, Tiantian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Ru Zhang, Xin Liu, Mingxuan Wang, Yonghui Wu, and Lin Yan. VAPO: efficient and reliable reinforcement learning for advanced reasoning tasks. *CoRR*, abs/2504.05118, 2025. doi: 10.48550/ARXIV.2504.05118. URL https://doi.org/10.48550/arXiv.2504.05118.

Ziqian Zeng, Jianwei Wang, Junyao Yang, Zhengdong Lu, Haoran Li, Huiping Zhuang, and Cen Chen. Privacyrestore: Privacy-preserving inference in large language models via privacy removal and restoration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 10821–10855. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.532/.

Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-r1: Teaching llms multi-round routing and aggregation via reinforcement learning. *CoRR*, abs/2506.09033, 2025. doi: 10.48550/ARXIV.2506.09033. URL https://doi.org/10.48550/arXiv.2506.09033.

Kaiyan Zhang, Jianyu Wang, Ning Ding, Biqing Qi, Ermo Hua, Xingtai Lv, and Bowen Zhou. Fast and slow generating: An empirical study on large and small language models collaborative decoding. *CoRR*, abs/2406.12295, 2024a. doi: 10.48550/ARXIV.2406.12295. URL https://doi.org/10.48550/arXiv.2406.12295.

Kaiyan Zhang, Jianyu Wang, Ermo Hua, Biqing Qi, Ning Ding, and Bowen Zhou. Cogenesis: A framework collaborating large and small language models for secure context-aware instruction following. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 4295–4312. Association for Computational Linguistics, 2024b. doi: 10.18653/V1/2024.ACL-LONG.235. URL https://doi.org/10.18653/v1/2024.acl-long.235.

Rui Zhang and Joel R. Tetreault. This email could save your life: Introducing the task of email subject line generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 1: Long Papers*, pp. 446–456. Association for Computational Linguistics, 2019.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Xinyu Zhang, Huiyu Xu, Zhongjie Ba, Zhibo Wang, Yuan Hong, Jian Liu, Zhan Qin, and Kui Ren. Privacyasst: Safeguarding user privacy in tool-using large language model agents. *IEEE Trans. Dependable Secur. Comput.*, 21(6):5242–5258, 2024c. doi: 10.1109/TDSC.2024.3372777. URL https://doi.org/10.1109/TDSC.2024.3372777.

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 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Banghua Zhu, Ying Sheng, Lianmin Zheng, Clark W. Barrett, Michael I. Jordan, and Jiantao Jiao. Towards optimal caching and model selection for large model inference. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

APPENDIX CONTENTS

A Limitations and Social Impact **17**

A.1 Limitations 17

A.2 Broader Social Impact 17

B Comparison with Existing Work **17**

B.1 Related Paradigms 17

B.2 Methodological Deltas from Prior Local-First Collaboration 18

C Privacy Injection Pipeline and PrivQA Dataset Details **18**

C.1 Privacy Injection Pipeline 18

C.2 PrivQA Dataset Details 19

D Dataset Synthesis, Training, and Inference Cost **19**

D.1 LLM Usage 19

D.2 Computational Overhead of Training 19

D.3 Computational Burden of Inference 20

E Format of SLM Trajectory **20**

F Human and Cross-Model Evaluation **21**

F.1 Accuracy of Privacy Leakage Judgment of the LLM 21

F.2 Quality of Privacy QA Synthesis Data 21

F.3 Agreement across Evaluator Model Families 22

G Evaluation metrics **22**

H Proof of Robustness to Evaluator Noise **24**

I Additional Experiment Details and Results **25**

I.1 Additional Training Details 25

I.2 Gains of Dynamic Collaboration 26

I.3 Mitigating Confident Hallucination on Hard Queries 27

I.4 Generalization to Real-World Personal Tasks 27

I.5 Impact of Different SLM 28

I.6 Impact of Different LLM 28

I.7 Ablation Experiment 29

I.8 Case Study 29

I.9 Frequency of LLM's two Feedback Actions 30

J Prompts for Reproduction **30**

A Limitations and Social Impact

A.1 Limitations

The RL-based strategy learning in our framework assumes basic instruction-following ability. Thus, extremely lightweight models may need a supervised cold-start to stabilize training and leverage LLM feedback effectively. In addition, although we formalize privacy leakage as an item-set accounting problem in Appendix G and empirically reduce privacy leakage to near zero on both PrivQA and real-world benchmarks, our framework only guarantees a substantial reduction rather than the complete elimination of privacy leakage.

A.2 Broader Social Impact

This work targets a socially beneficial direction: enabling capable personal AI while reducing the amount of user data disclosed to cloud services. By keeping user-attributed information on-device and forwarding only user-agnostic information queries, our framework can lower aggregate data exposure in commercial personal-intelligence deployments, e.g., on-device assistants that route through Private Cloud Compute or similar backends. It also provides infrastructure for privacy-aware research: we will release PrivQA and code with privacy items, injected into publicly available QA corpora, and no real user data is collected or released. As with any privacy technology, deployments should combine our framework with system-level access controls and standard LLM safety measures.

B Comparison with Existing Work

Table 6: Comparative overview of existing work and our dynamic collaboration framework.

Domain	Method	SLM Decision Space	Role of LLM	Privacy	Efficiency	Task Performance
Cloud-First Collaboration	Search-R1 (Jin et al., 2025)	SLM as a tool	tool-augmented LLM	✗	✗	✓
	RouterDC (Chen et al., 2024a)	candidate model	candidate answer model	✗	✗	✓
	RouterLLM (Ong et al., 2025)	model selection	candidate answer model	✗	✗	✓
Local-First Collaboration	AutoMix (Aggarwal et al., 2024)	forward unresolved query	passive tool	✗	✓	✓
	LLM Cascade (Yue et al., 2024)	cascade selection	passive tool	✗	✓	✓
	CoGenesis (Zhang et al., 2024b)	rewrite	reasoning tool	✓	✗	✓
	PAPILLON (Li et al., 2025)	rewrite + post-process	passive tool	✓	✗	✓
	PrivacyRestore (Zeng et al., 2025)	removing privacy + restoration	reasoning tool	✓	✗	✓
	Self-Disclosures (Dou et al., 2024)	PII masking	passive tool	✓	✗	✓
	Router-R1 (Zhang et al., 2025)	model selection	candidate answer model	✗	✗	✓
	Privacy-R1 (Hui et al., 2026)	chunk selection	passive tool	✓	✗	✓
Multi-Agent System	ConsensusLLM (Chen et al., 2023)	–	peer agent	✗	✗	✓
	MetaGPT (Huang et al., 2024)	–	peer agent	✗	✗	✓
	AgentVerse (Chen et al., 2024b)	–	peer agent	✗	✗	✓
	RoCo (Mandi et al., 2024)	–	peer agent	✗	✗	✓
	MrlX-DeepResearch (Hong et al., 2025)	–	peer research agent	✗	✗	✓
Ours	Dynamic Collaboration Framework	coordinate collaboration	collaborator with active feedback	✓	✓	✓

B.1 Related Paradigms

Table 6 summarizes representative prior work across five dimensions: SLM decision space, role of LLM, privacy preservation, interaction efficiency, and task performance. *Cloud-first* methods (Search-R1 (Jin et al., 2025), RouterDC (Chen et al., 2024a), RouterLLM (Ong et al., 2025)) treat the LLM as the primary inference engine and use SLMs mainly for auxiliary routing or tool use, thus inheriting the privacy and efficiency concerns of cloud-based LLMs. *Local-first* methods place the SLM on the user side but retain static interaction pipelines, which we elaborate in Appendix B.2 below. *Multi-agent systems* (MetaGPT (Hong et al., 2024), AgentVerse (Chen et al., 2024b), MrlX-DeepResearch (Hong et al., 2025)) emphasize communication and role specialization but typically assume shared task information and symmetric environment access; our setting is instead heterogeneous and information-asymmetric, with the LLM accessing only what the SLM selectively discloses.

B.2 Methodological Deltas from Prior Local-First Collaboration

Beyond the framework-level checkmarks in Table 6, three methodological axes distinguish our framework from prior local-first work.

SLM decision space. Existing methods constrain the SLM to a fixed action shape: confidence-based deferral (AutoMix (Aggarwal et al., 2024), LLM Cascade (Yue et al., 2024)), one-shot rewrite (PAPILLON (Li et al., 2025), PrivacyRestore (Zeng et al., 2025), CoGenesis (Zhang et al., 2024b)), token-level masking (Self-Disclosures (Dou et al., 2024)), single-shot routing among LLMs (Router-R1 (Zhang et al., 2025)), or chunk-level dispatch (Privacy-R1 (Hui et al., 2026)). Our SLM decides both *when* to consult and *how* to formulate each request, iteratively across turns, so the interaction pattern adapts to each query rather than following a fixed template.

Role of the LLM. Prior local-first work treats the LLM as a passive answer engine with no channel for signaling under-specified requests. We instead let the LLM choose between clarification and answering at each turn, turning it into an active collaborator whose feedback also serves as a process-level supervision signal for the SLM.

Optimization objective. Each prior method targets a subset of $\{\text{quality}, \text{efficiency}, \text{privacy}\}$: confidence-based routing trades quality for efficiency; rewriting or masking methods target privacy but do not model interaction cost; RL-based routers optimize task performance while relying on structural constraints for privacy. Our reward integrates all three objectives with tunable trade-off weights and is optimized end-to-end.

These deltas are complementary rather than independent: iterative when and how decisions require an active LLM to correct under-specified requests, and jointly optimizing all three objectives is what drives the SLM to prefer request reformulation over verbatim forwarding.

C Privacy Injection Pipeline and PrivQA Dataset Details

C.1 Privacy Injection Pipeline

We build PrivQA on question answering because this format is naturally suited to outcome-based RL, where the correctness of an answer provides a verifiable reward signal for learning the collaboration policy. As public QA corpora contain no user-attributed private content, we introduce privacy semantics in a controllable manner through an injection pipeline. We prompt an LLM to augment each existing QA pair through a three-step process (Figure 9), which we illustrate below with the seed pair Question: "Where does Kevin's family go in Home Alone 2?", Answer: "Florida."

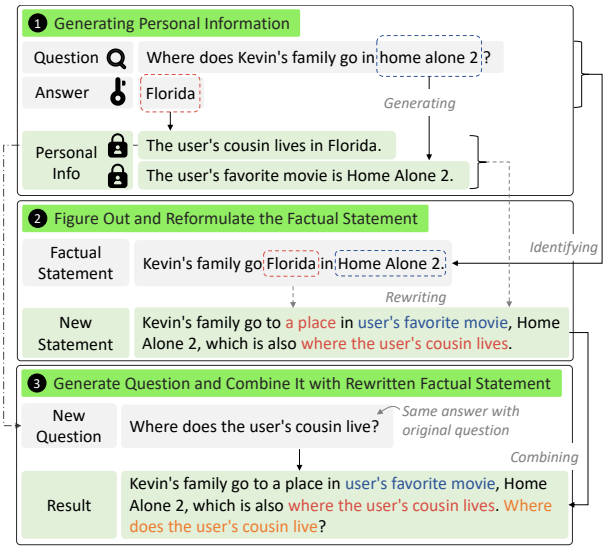


Figure 9: Privacy injection pipeline for PrivQA dataset processing.

Step 1: Personal-information generation. We first generate logically coherent personal information linked to the core entities in the QA pair—in this example, facts such as "The user's cousin lives in Florida" and "The user's favorite movie is Home Alone 2."

Step 2: Indirect reformulation. We identify the core factual statement ("Kevin's family go to Florida in Home Alone 2.") and reformulate it so the answer entity "Florida" is referenced indirectly through the generated personal information, producing a new statement like "Kevin's family go to a place in user's favorite movie, Home Alone 2, which is also where the user's cousin lives."

Step 3: Privacy-infused question synthesis. We synthesize a new question based on the personal information, "Where does the user's cousin live?" and combine it with the reformulated statement to create a privacy-infused QA pair: "Kevin's family go to a place in user's favorite movie, Home Alone 2, which is also where the user's cousin lives. Where does the user's cousin live?" The answer remains "Florida," but obtaining it now requires inference over embedded personal information, effectively simulating real-world scenarios where users might unwittingly disclose private details.

Step 4: Validation. After generation, we apply an LLM-based filter to remove cases where (1) the answer is altered or (2) the answer is explicitly embedded in the question. Human evaluation on 200 randomly sampled instances further confirms that 99% are correctly reformulated with accurate answers and appropriate privacy infusion (see Appendix F).

C.2 PrivQA Dataset Details

Table 7: Dataset Statistics. NQ and HotpotQA are in-domain datasets with training set, while TriviaQA, PopQA, 2Wiki, and MuSiQuE are out-domain datasets. Out-domain datasets have no training data (denoted by "–") and are evaluated on the test set only.

Type	Dataset	Training Set			Test Set		
		# Sample	# Privacy Info	Avg Input Token	# Sample	# Privacy Info	Avg Input Token
General	NQ (Kwiatkowski et al., 2019)	54,589	151,128	44.09	2,014	5,778	46.01
	TriviaQA (Joshi et al., 2017)	–	–	–	5,057	16,582	53.05
	PopQA (Mallen et al., 2022)	–	–	–	10,448	21,940	35.56
Multi-Hop	HotpotQA (Yang et al., 2018)	65,812	228,392	55.19	5,148	17,040	53.53
	2Wiki (Ho et al., 2020)	–	–	–	8,463	23,117	44.72
	MuSiQuE (Trivedi et al., 2022)	–	–	–	1,956	6,788	55.92

We construct the Privacy QA (PrivQA) dataset by augmenting six public QA datasets with a privacy injection pipeline. To thoroughly evaluate the generalization capability of the learned collaboration strategy, we construct the training set by including only samples derived from NQ and HotpotQA, while the test set covers all six datasets, including NQ, TriviaQA, PopQA, HotpotQA, 2Wiki, and MuSiQuE. As summarized in Table 7, the in-domain training set contains a total of 120K samples with an average input token length of 50.16. The out-domain test sets are also diverse, spanning a range of domains and question complexities, with input lengths varying from 35.56 to 55.92 tokens on average. Furthermore, multi-hop datasets tend to have longer inputs, reflecting their compositional reasoning requirements.

D Dataset Synthesis, Training, and Inference Cost

D.1 LLM Usage

We use Qwen3-Max, a closed-source LLM with strong instruction-following ability, for our PrivQA dataset synthesis. The total token consumption is 55.6M tokens for input and 30.9M tokens for output. The total annotation cost amounts to approximately $65.59 and the average cost is $0.0005 per sample. The main LLM used in the collaboration experiment, Qwen3-235B-A22B-Instruct, is deployed on a local server equipped with eight H20 GPUs, and therefore does not incur any API costs.

D.2 Computational Overhead of Training

Our RL-based training incurs approximately 190 GPU-hours for Qwen3-8B (about 24 hours of wall-clock time on 8×H20 GPUs) and 110 GPU-hours for Qwen3-4B (about 14 hours on 8×H20 GPUs). This cost is a one-time offline training investment, and no training is performed on the device. Table 8 compares learning-based methods trained on the same base SLM and PrivQA training data. Under matched GPU-hours, our method with Qwen3-8B outperforms SFT with Qwen3-8B by 15.3% average EM, and our method with Qwen3-4B (110 GPU-hours) already surpasses SFT with Qwen3-8B (140 GPU-hours) despite using less compute. This confirms that the improvement stems from the collaboration design rather than additional supervision or compute.

Table 8: Training cost and average EM of learning-based methods on 8×H20 GPUs, where all methods share the same base SLM and the same PrivQA training data.

Method	Training	GPU-hours	Avg EM
SFT (Qwen3-8B)	SFT	140	27.80
Router-R1 (Qwen3-8B)	RL	190	37.96
Ours (Qwen3-4B)	RL	110	38.34
Ours (Qwen3-8B)	RL	190	43.11

D.3 Computational Burden of Inference

Table 9: Average performance and latency per sample of different methods over PrivQA datasets.

Method	Avg EM	Avg Latency Per Sample (ms)
Edge SLM Deployment	25.77	243
Cloud-side LLM Inference	47.09	1096
Ours	46.01	675

Latency. We report the average per-sample latency for edge SLM deployment, cloud-side LLM inference, and SLM-LLM collaboration in Table 9. The SLM, Qwen3-8B, is deployed locally on a single NVIDIA 4090 GPU, while the LLM, Qwen3-Max, is accessed via a cloud API. Our collaborative approach delivers performance close to cloud LLM inference (46.01 vs. 47.09 EM) while cutting latency by 38% (675 vs. 1096 ms), at the cost of $\sim$2.8$\times$ higher latency than local-only edge SLM inference. The worst-case tail latency reaches 2283 ms in Table 11, but is triggered only for the 0.09% of samples that require multi-turn interaction; the remaining 99.9% of samples stay well within a typical interactive envelope.

Table 10: Upstream and downstream payload per sample and average end-to-end latency of different methods, where the SLM is Qwen3-8B.

Method	Upstream Token	Downstream Token	Avg Latency
LLM Inference	47.68	251.12	1096 ms
Self-Disclosures (Dou et al., 2024)	44.91	264.50	1100 ms
PAPILLON (Li et al., 2025)	28.60	192.11	1108 ms
AutoMix (Aggarwal et al., 2024)	21.35	305.22	428 ms
Router-R1 (Zhang et al., 2025)	25.18	158.60	688 ms
Ours	20.65	161.97	675 ms

Bandwidth, cost, and energy. We further measure the communication volume between the device and the cloud, the monetary cost of the cloud API, and the energy consumed on the device under the same setting. For each method, we tokenize the upstream payload (tokens sent to the cloud LLM) and the downstream payload (tokens returned by the LLM), and report them together with the latency in Table 10. Our method uploads only 20.65 tokens per sample, compared with 47.68 tokens when the raw user query is sent directly to the cloud, a 57% reduction in upstream bandwidth. Furthermore, on the device side, the RTX 4090 with a TDP of 450 W running for 675 ms consumes only 0.084 Wh per sample.

Multi-turn distribution. Since multi-turn interactions could in principle delay the response, we further report the distribution of interaction turns and the tail latency on each PrivQA test set in Table 11. Although the interaction is capped at five turns, only 0.85 turns are used per sample on average, and the interaction terminates within a single turn for 99% of the samples, with the largest observed turn count being three. The tail latency stays bounded accordingly.

E Format of SLM Trajectory

Figure 10 shows examples with correct and incorrect formats. The generated trajectory of the SLM should start with "<think> ... </think>" and end with "<answer> ... </answer>" block. When SLM wants to request LLM, it should put its request within "<message> ... </message>" block

Table 11: Distribution of interaction turns and tail latency of our method on each PrivQA test set, where * represents out-domain datasets. The last row reports the average over the six datasets for the turn distribution and the maximum over all samples for the latency.

Dataset	0-turn (%)	1-turn (%)	≥ 2-turn (%)	Max Latency
NQ	12.8	87.1	0.10	1459 ms
TriviaQA*	20.5	79.5	0.04	1123 ms
PopQA*	7.4	92.5	0.09	1872 ms
HotpotQA	10.4	89.5	0.10	1245 ms
2Wiki*	5.0	94.8	0.21	2283 ms
MuSiQuE*	12.5	87.5	0.00	761 ms
Overall	11.4	88.5	0.09	2283 ms

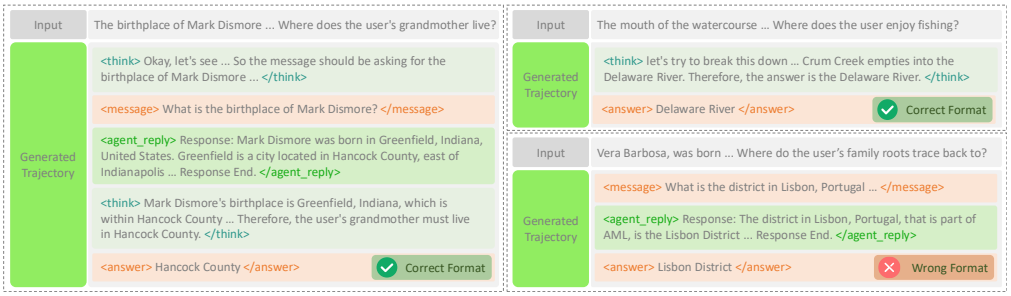


Figure 10: Correct and incorrect cases of format of SLM trajectories.

and it will receive the LLM's feedback within "<agent_reply> ... </agent_reply>". There must be exactly one think block preceding any message block. Moreover, a trajectory may contain no message block at all, indicating that the SLM has determined no request is necessary.

F Human and Cross-Model Evaluation

F.1 Accuracy of Privacy Leakage Judgment of the LLM

To evaluate the accuracy of the privacy leakage judgments made by the LLM, we conduct a human evaluation involving three independent annotators from the laboratory. A random sample of 200 instances is selected, and each annotator independently assesses whether the request generated by the SLM associated with each sample contained a disclosure of the given user's private information. The annotators assign binary labels to each instance, marking it as "true" if privacy leakage was present and "false" if there was no privacy leakage. The final ground truth labels for each sample are determined using a majority-voting strategy among the three annotators. Based on this annotated ground truth, we evaluated the performance of the LLM's judgments by calculating its accuracy, which reached 98.5%. We find that the LLM can reliably perform binary classification on whether a request leaks labeled private information.

F.2 Quality of Privacy QA Synthesis Data

To assess the quality of the generated privacy-infused QA data, we also conduct a human evaluation with three independent annotators to evaluate a randomly selected set of samples. The annotators follow two main criteria: (1) whether the generated questions containing privacy information are semantically fluent, and (2) whether the answer to the generated question is consistent with the answer to the original question. Each sample is labeled as "true" if it meets both criteria, and "false" otherwise, following a binary classification approach. The final label for each sample is determined by majority voting among the annotators. The proportion of samples labeled "true" is the overall quality score. Through this process, we find that 99% of the generated data are correctly reformulated with accurate answers and appropriate privacy information infusion.

F.3 Agreement across Evaluator Model Families

To assess the robustness of our privacy evaluation to the choice of evaluator model, we replace the evaluator with LLMs from different families, including Qwen3-235B-A22B-Instruct, Qwen3-Max, DeepSeek-R1, and DeepSeek-V3, and let each independently judge whether the SLM's requests leak privacy. Their pairwise sample-level agreement is reported in Table 12.

Table 12: Pairwise sample-level agreement (%) among four evaluators from different model families.

Evaluator	Qwen3-235B	Qwen3-Max	DeepSeek-R1	DeepSeek-V3
Qwen3-235B	–	98.17	97.89	97.94
Qwen3-Max	98.17	–	97.87	97.42
DeepSeek-R1	97.89	97.87	–	97.69
DeepSeek-V3	97.94	97.42	97.69	–

All pairwise agreements fall within a narrow band of 97.42% – 98.17%, and the cross-family agreements between the Qwen and DeepSeek models range from 97.42% to 97.94%. Since the DeepSeek family is involved in neither the data synthesis nor the training pipeline, its consistent agreement confirms that the near-zero leakage of our method is not an artifact of same-family bias between the injection and evaluation LLMs, and that the privacy evaluation is robust to the choice of evaluator model.

G Evaluation metrics

As aligned with our objectives, we design different evaluation metrics to assess (1) the response quality of the SLM, (2) the interaction efficiency between the SLM and the LLM, and (3) the degree of privacy leakage.

Performance. We evaluate model performance using Exact Match (EM), a widely adopted metric in question answering (QA) tasks. EM measures the percentage of predictions that are identical to the ground-truth answers. Formally, given a predicted answer s_{pred} and the ground-truth answer s_{gt}, the EM score is defined as:

$$\text{EM} = \begin{cases} 1, & \text{if } s_{\text{pred}} = s_{\text{gt}} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The dataset-level EM is the average over all samples: $\text{EM}_{\text{dataset}} = \frac{1}{N}\sum_{i=1}^{N} \text{EM}(s^i_{\text{pred}}, s^i_{\text{gt}})$. EM requires perfect string-level agreement between s_{pred} and s_{gt}. To complement EM and better capture semantic similarity between the predicted and ground-truth answers, we also employ BERTScore (Zhang et al., 2020). For each answer pair, BERTScore computes a similarity score based on contextual embeddings. The overall BERTScore for the dataset, denoted as F_{BERT}, is calculated by averaging the individual BERTScore values:

$$F_{\text{BERT}} = \frac{1}{N}\sum_{i=1}^{N} \text{BERTScore}(s^i_{\text{pred}}, s^i_{\text{gt}}) \tag{2}$$

where N is the total number of samples. For the real-world personal-task benchmarks, including PUPA, AESLC, and SAMSum, which are open-ended generation tasks without a unique gold answer, we adopt LLM-as-a-judge with prompted rubrics (Zheng et al., 2023). The specific quality-preservation protocol for PUPA and the five-dimension rubric for AESLC/SAMSum are detailed in Appendix I.4.

Efficiency. We measure the interaction efficiency between the SLM and the LLM by calculating the average number of interaction rounds per sample. A lower average indicates that the SLM is able to independently answer more queries or, when necessary, issue fewer redundant or low-quality requests for assistance.

In addition, we introduce the Interaction Necessity Score (INScore), an efficiency metric calibrated by query difficulty. We divide queries into easy or hard based on whether the SLM can answer without assistance. The ideal policy is for the SLM to request assistance only when it cannot answer

a query without assistance. We formalize this as a classification problem, where query difficulty serves as the ground-truth label ("easy" or "hard"), and the decision to request LLM assistance is treated as the model's prediction. The INScore is computed as the average of two recall values: (1) the recall for easy queries, reflecting the proportion of easy queries for which the SLM does not request assistance, defined as

$$\text{Recall}_{\text{easy}} = \frac{|Q_{\text{w/o req}} \cap Q_{\text{easy}}|}{|Q_{\text{easy}}|} \tag{3}$$

where Q_{easy} denotes the set of easy queries, $Q_{\text{w/o req}}$ denotes the set of queries for which the SLM does not issue requests, and (2) the recall for hard queries, reflecting the proportion of hard queries for which the SLM does request assistance, defined as

$$\text{Recall}_{\text{hard}} = \frac{|Q_{\text{hard}} \cap Q_{\text{w/ req}}|}{|Q_{\text{hard}}|} \tag{4}$$

where Q_{hard} denotes the set of hard queries and $Q_{\text{w/ req}}$ denotes the set of queries for which the SLM does make a request. Then, the INScore is defined as

$$\text{INScore} = \frac{\text{Recall}_{\text{easy}} + \text{Recall}_{\text{hard}}}{2} \tag{5}$$

In the ideal case where the SLM never requests help for easy queries and always requests help for hard queries, INScore achieves its maximum value of 1.0. By contrast, in two degenerate cases where the SLM always requests help or never requests help (even on hard queries), INScore drops to 0.5. By defining easy and hard relative to each SLM's own standalone capability, INScore serves as a self-calibrated efficiency metric that assesses whether a policy issues requests exactly when its own SLM cannot answer independently.

Privacy. Formal ε-differential-privacy (DP) guarantees are not directly available for open-ended natural-language interaction, since DP mechanisms rely on bounded sensitivity over a fixed output space that free-form generation lacks (Dwork et al., 2006). Privacy in LLM interaction is therefore commonly quantified by item-set based leakage accounting (Mireshghallah et al., 2024; Li et al., 2025), aligning with inferential-privacy notions in the spirit of Pufferfish privacy (Kifer & Machanavajjhala, 2014): rather than bounding output stability, we bound what an honest-but-curious LLM adversary[1], observing only the SLM's requests rather than the user's original query, can additionally infer about the user. Concretely, we measure leakage as the increase in the LLM's knowledge caused by these requests, and consider a piece of information privacy-sensitive if its type belongs to a predefined set (Ignatius & Kokkonen, 2007; Dou et al., 2024). Under this threat model, the privacy leakage induced by a multi-turn request set is defined as follows.

Definition 1 (Privacy Leakage in SLM-LLM Interaction) *Let K_{LLM} denote the prior knowledge of the LLM about the user, which contains no privacy item, $R = \{r_1, \ldots, r_T\}$ the multi-turn request set issued by the SLM, and $Priv_{item}$ the set of privacy-sensitive item types. The privacy leakage of R is*

$$Leak(R) = \big\{I \mid I \in K_{LLM} \cup R,\ Type(I) \in Priv_{item}\big\} \setminus \big\{I \mid I \in K_{LLM},\ Type(I) \in Priv_{item}\big\}.$$

By construction, $\text{Leak}(R) = \varnothing$ certifies that no new privacy item becomes derivable from R relative to K_{LLM}, so under a randomized SLM policy $M : Q \mapsto R$ the adversary's posterior over the user's privacy items equals its prior. Consequently, $\Pr[\text{Leak}(R) \neq \varnothing]$ upper-bounds the fraction of samples on which the adversary can gain any inferential advantage, up to the false-negative rate of the LLM judge that operationalizes $\text{Leak}(\cdot)$ (bounded at $\sim 1.5\%$ by human agreement in Appendix F). Our empirical estimator $\text{Priv}_{\text{sample}}$, as defined below, can therefore be read as an empirical upper bound on this per-sample inferential advantage, rather than as a formal ε-guarantee.

Specifically, we evaluate the degree of privacy leakage in the SLM's request sent to the LLM using two metrics that instantiate $\text{Leak}(R)$ on each sample. Both metrics operate at the semantic level, so that paraphrases or generalized references to a private item are captured in the same way as verbatim

[1]Threats outside this model are orthogonal to our accounting: adversarial LLMs (e.g., probing via clarifications) are partially penalized by the evaluator's trajectory-level scoring at training time, with stronger defenses requiring independent trust layers; prompt injection via cited content and side channels (timing, request counts) fall to standard LLM safety and system-level mechanisms.

mentions. First, we compute the average semantic similarity between the uploaded request (req) and the set of privacy information $P = \{p_1, \ldots, p_m\}$ labeled from the original data (Ngong et al., 2025), where m denotes the number of privacy items, i.e., the user-related items whose types fall in $\text{Priv}_{\text{item}}$. Specifically, the privacy leakage score based on BERTScore is defined as:

$$\text{Priv}_{\text{sim}} = \frac{1}{|P|} \sum_{i=1}^{m} \text{BERTScore}(\text{req}, p_i). \tag{6}$$

A lower score indicates less semantic overlap and thus lower privacy leakage. Second, we measure the proportion of samples where the SLM issued at least one privacy-leaking request ($\text{Priv}_{\text{sample}}$) with LLM judge (Zheng et al., 2023; Li et al., 2025). Specifically, given the reference set P and the requests of the SLM concatenated across all interaction turns, the LLM judges whether they allow inference of any user-related private information, i.e., whether $\text{Leak}(R) \neq \varnothing$. Since the judge observes the whole request set jointly rather than each request in isolation, it identifies leaks whether the privacy item is disclosed in a single request or is only recoverable by combining several. In PrivQA, the multi-hop subsets (HotpotQA, 2WikiMultihopQA, MuSiQuE) naturally exercise the latter, since answering their queries already requires composing multiple facts. The reported $\text{Priv}_{\text{sample}}$ is thus measured at the level of the whole request set. For PrivQA, P is given by the privacy items injected during dataset synthesis as logically consistent indirect references produced by the pipeline of Appendix C, so a leak requires inferring the private item semantically from paraphrased content rather than from verbatim disclosure. For real-world user data, P is obtained from dedicated privacy-detection tools.

H Proof of Robustness to Evaluator Noise

Because the privacy reward r_{privacy} is produced by an LLM judge rather than an exact verifier, we bound the effect of the judge's errors on policy optimization. Prior work (Wang et al., 2020) shows that a policy trained on a bounded discrete reward corrupted by iid Bernoulli noise converges to the same optimum as under the noiseless reward, with only the convergence speed affected. We adapt this result to our multi-term reward structure in Lemma 1, and further generalize it in Lemma 2 to the case where the flip rates may be asymmetric between false negatives and false positives and vary across inputs.

The privacy penalty $r_{\text{privacy}} \in \{-1, 0\}$ is a sparse binary signal and the only reward term that depends on the judgment of the evaluator LLM, while the format, quality, feedback, and efficiency terms are computed by exact rules. Grouping the exactly computed terms into $C = r_{\text{format}} + r_{\text{quality}} + r_{\text{feedback}} + \beta r_{\text{efficiency}}$, the reward function of Section 4.4 becomes $R = C + \alpha r_{\text{privacy}}$.

Lemma 1 (Direction preservation under symmetric iid Bernoulli evaluator noise) *Assume the evaluator LLM flips its privacy judgment symmetrically and independently with a single probability $p < 1/2$ that does not depend on the input. Then the expected policy gradient under the noisy reward $\hat{R}$ has the same direction as under the noiseless reward R, with the privacy component scaled by a positive factor $(1 - 2p)$.*

Proof If the evaluator flips its judgment independently with probability p, the observed penalty relates to the true one as

$$\hat{r}_{\text{privacy}} = \begin{cases} r_{\text{privacy}}, \text{ with probability } 1 - p \\ -1 - r_{\text{privacy}}, \text{ with probability } p \end{cases}$$

so that a leaking request is missed and a safe request is falsely penalized with the same probability, and its expectation is

$$\mathbb{E}[\hat{r}_{\text{privacy}}] = (1 - p) r_{\text{privacy}} + p(-1 - r_{\text{privacy}}) = (1 - 2p) r_{\text{privacy}} - p.$$

The value model is trained on the observed reward $\hat{R} = C + \alpha \hat{r}_{\text{privacy}}$ and hence converges to $V(s_0) = \mathbb{E}[\hat{R} \mid s_0]$ for a user query s_0. For a rollout i, the trajectory-level advantage $\hat{A}_i = \hat{R}_i - V(s_0)$ then satisfies

$$\mathbb{E}[\hat{A}_i] = \big(C_i - \mathbb{E}[C \mid s_0]\big) + \alpha(1 - 2p)\big(r_{\text{privacy},i} - \mathbb{E}[r_{\text{privacy}} \mid s_0]\big),$$

where the additive bias $-\alpha p$ appears in both $\hat{R}_i$ and $V(s_0)$ and therefore cancels in the advantage; only the reward-magnitude-dependent term $\alpha(1-2p)(r_{\text{privacy},i} - \mathbb{E}[r_{\text{privacy}}|s_0])$ remains. Compared with the noiseless advantage recovered at $p = 0$, evaluator noise leaves the exactly computed terms untouched and scales the privacy component by $(1-2p)$, which is positive whenever $p < 1/2$. Since $\nabla_\theta J = \mathbb{E}[\hat{A}\,\nabla_\theta \log \pi_\theta]$, the noise scales the privacy component of $\nabla_\theta J$ by a positive factor $(1-2p)$ but preserves its sign. □

Lemma 1 assumes that flipping is symmetric between false negatives and false positives and that the flip rate is a single constant p independent of the input. We now show that direction preservation extends to the more general case where both assumptions are relaxed.

Lemma 2 (Direction preservation under asymmetric and heterogeneous evaluator noise) *Let the evaluator's false-negative and false-positive rates $p_{\text{FN}}(x), p_{\text{FP}}(x)$ depend on the input x, with mean combined error $\bar{q} = \mathbb{E}_x[p_{\text{FN}}(x) + p_{\text{FP}}(x)] < 1$ and variance $\sigma^2 = \text{Var}_x(p_{\text{FN}}(x)) + \text{Var}_x(p_{\text{FP}}(x))$. Then the expected policy gradient under the noisy reward $\hat{R}$ decomposes as*

$$\nabla_\theta J_{\hat{R}} \;=\; \nabla_\theta J_R^{\text{exact}} + (1-\bar{q})\,\nabla_\theta J_R^{\text{priv}} + \varepsilon, \qquad \|\varepsilon\|_2 \;\le\; (1+\sqrt{2})\,\alpha\,\sigma\sqrt{\mathbb{E}_x\|\nabla_\theta \log \pi_\theta(\cdot \mid x)\|^2},$$

where $\nabla_\theta J_R^{\text{exact}}$ groups the exactly-computed reward terms and $\nabla_\theta J_R^{\text{priv}}$ is the privacy-component gradient under the noiseless reward.

Proof Let $y = -r_{\text{privacy}} \in \{0,1\}$ be the leak indicator and $q(x) = p_{\text{FN}}(x) + p_{\text{FP}}(x)$. Under per-input flip rates, a direct calculation gives

$$\mathbb{E}[\hat{r}_{\text{privacy}} \mid r_{\text{privacy}}, x] \;=\; -p_{\text{FP}}(x) + (1-q(x))\,r_{\text{privacy}}.$$

Writing $\Delta_{\text{FP}}(x) = p_{\text{FP}}(x) - \bar{p}_{\text{FP}}$ and $\Delta_q(x) = q(x) - \bar{q}$, expanding the trajectory-level advantage $\hat{A}_i = \hat{R}_i - V(s_0)$ with $V(s_0) = \mathbb{E}_x[\hat{R} \mid s_0]$, and applying $\nabla_\theta J = \mathbb{E}_x[\hat{A}\,\nabla_\theta \log \pi_\theta]$ (constants vanish since $\mathbb{E}_x[\nabla_\theta \log \pi_\theta] = 0$), the noisy gradient becomes

$$\begin{aligned}\nabla_\theta J_{\hat{R}} = \nabla_\theta J_R^{\text{exact}} &+ (1-\bar{q})\,\nabla_\theta J_R^{\text{priv}} \\ &- \alpha\,\mathbb{E}_x[\Delta_{\text{FP}}(x)\,\nabla_\theta \log \pi_\theta] - \alpha\,\mathbb{E}_x[\Delta_q(x)\,r_{\text{privacy}}(x)\,\nabla_\theta \log \pi_\theta].\end{aligned}$$

Denote the last two terms by ε. Applying Cauchy–Schwarz componentwise, using $|r_{\text{privacy}}| \le 1$, and bounding $\text{Var}(q) \le 2\sigma^2$ via $2\,\text{Cov}(p_{\text{FN}}, p_{\text{FP}}) \le \text{Var}(p_{\text{FN}}) + \text{Var}(p_{\text{FP}})$,

$$\|\varepsilon\|_2 \;\le\; \alpha\Big(\sqrt{\text{Var}(p_{\text{FP}})} + \sqrt{2}\,\sigma\Big)\sqrt{\mathbb{E}_x\|\nabla_\theta \log \pi_\theta\|^2} \;\le\; (1+\sqrt{2})\,\alpha\,\sigma\sqrt{\mathbb{E}_x\|\nabla_\theta \log \pi_\theta\|^2}.$$

Two special cases follow. (i) At $\sigma = 0$, $\varepsilon = 0$ and the direction of $\nabla_\theta J_{\hat{R}}$ is preserved exactly, with only a positive shrinkage $(1-\bar{q}) > 0$ on the privacy component. (ii) At $p_{\text{FN}}(x) = p_{\text{FP}}(x) = p$, we have $\bar{q} = 2p$, $\sigma = 0$, and Lemma 1 is recovered. □

Two independent measurements in our setting jointly constrain the assumptions of both lemmas. The human evaluation in Appendix F reports 98.5% agreement with the majority-vote annotations of three annotators, upper-bounding the disagreement rate at ≈ 0.015; interpreted under Lemma 1 this gives $p \le 0.015$, and interpreted under Lemma 2 it corresponds to $\bar{q} \lesssim 0.03$. In either reading, the dominant scaling remains at least 0.97 and the attenuation of the privacy signal is negligible. Cross-family agreement across four evaluators from the Qwen and DeepSeek families stays within a 97.42 – 98.17% band (Appendix F.3, Table 12), so no single family's systematic bias dominates; treating cross-family variance as an empirical proxy for the input-conditional variance σ places σ on the order of 10^{-2}, so the residual heterogeneity term ε contributes at most a few percent of the dominant scaled gradient. Consistent with these bounds, the training reward in Figure 5a increases monotonically without collapse.

I Additional Experiment Details and Results

I.1 Additional Training Details

We train with VAPO (Yue et al., 2025) for 600 steps at a batch size of 256 and a sampling temperature of 1.0. The maximum context length is 8,192 tokens for the local SLM and 16,384 tokens for the

cloud LLM. The actor and critic learning rates are 1e-6 and 2e-6, respectively, and we follow VAPO in setting the asymmetric PPO clip ratios to $\varepsilon_{\text{low}} = 0.2$ and $\varepsilon_{\text{high}} = 0.28$.

For a fair comparison, all baselines share the same SLM-LLM pair, maximum context length, and parameter dtype as our method. Method-specific settings, including SLM and LLM prompts, strictly follow the original papers (Aggarwal et al., 2024; Dou et al., 2024; Li et al., 2025; Zhang et al., 2025).

I.2 Gains of Dynamic Collaboration

Table 13: Performance of SLM with dynamic collaboration with Qwen3-235B-A22B-Instruct and baselines on different datasets.

Method	Model	General QA						Multi-Hop QA						Avg	
		NQ		TriviaQA*		PopQA*		HotpotQA		2Wiki*		MuSiQuE*			
		EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑
CoT	Qwen3-235B	40.91	75.89	66.30	85.95	45.44	73.08	30.81	69.26	47.91	76.81	23.62	66.23	42.50	74.54
SFT	Qwen3-4B	24.88	68.31	34.13	55.81	18.19	57.42	13.54	62.36	33.20	63.44	12.37	63.91	22.72	61.87
	Qwen3-8B	33.91	72.11	44.35	59.29	22.97	59.87	17.02	64.64	33.90	64.54	14.62	64.40	27.80	64.14
AutoMix Aggarwal et al. (2024)	Qwen3-4B	29.00	68.87	51.73	78.10	32.22	66.38	21.02	61.29	40.35	71.92	16.41	56.71	31.79	67.21
	Qwen3-8B	32.52	50.86	56.85	51.70	33.92	53.34	21.91	49.38	40.27	54.03	17.18	49.15	33.78	51.41
Self-Disclosures Dou et al. (2024)	Qwen3-4B	27.26	63.62	46.93	74.50	23.54	54.79	17.70	57.42	24.81	58.36	12.78	57.16	25.50	60.98
	Qwen3-8B	30.69	65.62	51.28	77.33	26.29	56.94	21.87	61.07	28.24	61.78	16.16	59.86	29.09	63.77
PAPILLON Li et al. (2025)	Qwen3-4B	32.77	68.03	54.78	79.27	36.73	65.80	22.38	61.38	33.03	65.91	15.64	58.61	32.55	66.50
	Qwen3-8B	35.40	69.66	56.36	80.39	35.88	64.78	23.82	62.55	31.47	64.74	16.46	58.32	33.23	66.74
Router-R1 Zhang et al. (2025)	Qwen3-4B	35.33	71.15	57.81	80.33	37.91	67.07	23.60	63.75	40.42	71.22	17.15	60.94	35.37	69.08
	Qwen3-8B	39.08	74.00	60.59	83.12	42.20	71.45	24.65	64.92	43.48	73.73	17.74	61.05	37.96	71.38
Ours	Qwen3-4B	39.42	76.43	61.82	83.95	44.17	74.08	25.51	67.30	40.52	74.33	18.61	65.00	38.34	73.51
	Qwen3-8B	**46.08**	**78.51**	**64.64**	**85.32**	**47.87**	**75.58**	**29.66**	**69.21**	**47.09**	**77.35**	**23.31**	**67.34**	**43.11**	**75.55**

Table 14: Privacy leakage of SLM with dynamic collaboration with Qwen3-235B-A22B-Instruct and static interaction framework on different datasets.

Method	Model	General QA						Multi-Hop QA						Avg	
		NQ		TriviaQA*		PopQA*		HotpotQA		2Wiki*		MuSiQuE*			
		Priv_{sim} ↓	$\text{Priv}_{\text{sample}}$ ↓	Priv_{sim} ↓	$\text{Priv}_{\text{sample}}$ ↓	Priv_{sim} ↓	$\text{Priv}_{\text{sample}}$ ↓	Priv_{sim} ↓	$\text{Priv}_{\text{sample}}$ ↓	Priv_{sim} ↓	$\text{Priv}_{\text{sample}}$ ↓	Priv_{sim} ↓	$\text{Priv}_{\text{sample}}$ ↓	Priv_{sim} ↓	$\text{Priv}_{\text{sample}}$ ↓
CoT	Qwen3-235B	66.53	100.00	64.40	100.00	68.34	100.00	64.34	100.00	67.96	100.00	64.50	100.00	66.01	100.00
SFT	Qwen3-4B	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Qwen3-8B	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
AutoMix Aggarwal et al. (2024)	Qwen3-4B	62.21	72.05	60.02	68.36	66.05	76.61	63.55	76.30	65.07	75.02	64.42	77.51	63.55	74.31
	Qwen3-8B	62.15	64.45	59.08	60.96	66.01	72.95	62.99	71.60	64.55	70.50	63.91	72.96	63.12	68.90
Self-Disclosures Dou et al. (2024)	Qwen3-4B	64.20	57.65	62.72	62.67	64.71	58.49	62.67	62.45	63.90	50.56	62.48	59.76	63.45	58.60
	Qwen3-8B	62.50	51.54	61.26	54.48	63.24	55.57	61.19	51.24	62.14	45.20	60.95	51.23	61.88	51.54
PAPILLON Li et al. (2025)	Qwen3-4B	62.89	30.14	61.50	37.87	63.23	36.97	61.86	32.96	63.85	25.82	62.19	32.11	62.58	32.64
	Qwen3-8B	60.57	22.99	58.81	29.84	60.00	18.80	60.07	28.73	62.58	18.69	60.29	27.20	60.39	24.37
Router-R1 Zhang et al. (2025)	Qwen3-4B	62.09	28.02	58.84	31.99	63.76	30.75	62.34	29.56	61.59	20.68	62.05	31.78	61.78	28.80
	Qwen3-8B	58.56	23.28	53.87	24.82	59.90	18.01	56.90	26.33	58.63	13.91	57.28	19.71	57.52	21.01
Ours	Qwen3-4B	47.80	0.149	42.03	0.356	50.92	0.048	48.38	0.350	54.61	0.225	46.56	0.409	48.38	0.256
	Qwen3-8B	**43.51**	**0.120**	**35.33**	**0.110**	**50.83**	**0.010**	**48.21**	**0.039**	**53.86**	**0.050**	**48.32**	**0.153**	**46.68**	**0.080**

Table 15: Efficiency performance of SLM with dynamic collaboration with Qwen3-235B-A22B-Instruct and baselines on different datasets.

Method	Model	General QA						Multi-Hop QA						Avg	
		NQ		TriviaQA*		PopQA*		HotpotQA		2Wiki*		MuSiQuE*			
		Avg Turn ↓	INScore ↑	Avg Turn ↓	INScore ↑	Avg Turn ↓	INScore ↑	Avg Turn ↓	INScore ↑	Avg Turn ↓	INScore ↑	Avg Turn ↓	INScore ↑	Avg Turn ↓	INScore ↑
CoT	Qwen3-235B	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00
SFT	Qwen3-4B	0.000	50.00	0.000	50.00	0.000	50.00	0.000	50.00	0.000	50.00	0.000	50.00	0.000	50.00
	Qwen3-8B	0.000	50.00	0.000	50.00	0.000	50.00	0.000	50.00	0.000	50.00	0.000	50.00	0.000	50.00
AutoMix Aggarwal et al. (2024)	Qwen3-4B	0.720	57.75	0.684	57.96	0.766	59.61	0.763	54.68	0.750	56.72	0.775	52.94	0.743	56.61
	Qwen3-8B	**0.644**	60.17	**0.610**	58.94	**0.730**	59.41	**0.716**	56.01	**0.705**	58.73	**0.730**	55.12	**0.689**	58.06
Self-Disclosures Dou et al. (2024)	Qwen3-4B	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00
	Qwen3-8B	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00
PAPILLON Li et al. (2025)	Qwen3-4B	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00
	Qwen3-8B	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00	1.000	50.00
Router-R1 Zhang et al. (2025)	Qwen3-4B	1.145	51.47	1.111	53.57	1.194	51.92	1.165	51.61	1.129	53.66	1.180	54.56	1.154	52.80
	Qwen3-8B	1.012	54.70	0.831	57.34	1.203	55.30	1.101	53.56	1.161	57.12	1.140	55.12	1.075	55.52
Ours	Qwen3-4B	0.873	60.50	0.796	61.93	0.927	**60.50**	0.897	**59.97**	0.953	53.81	0.875	60.73	0.887	59.57
	Qwen3-8B	0.788	**60.97**	0.668	**64.27**	0.926	60.23	0.891	57.91	0.920	**60.06**	0.898	**62.00**	0.849	**60.91**

Tables 13 - 15 provide the per-dataset breakdown of the aggregate results in Table 1 of the main text, along the three axes: quality, privacy, and efficiency.

Quality (Table 13). Our method is competitive with, and on several splits surpasses, the Qwen3-235B teacher: on NQ, Qwen3-8B reaches an EM of 46.08 versus the teacher's 40.91. Aggregated over the six datasets, our method improves average EM by 3.0% – 15.6% on Qwen3-4B and 5.2% – 15.3% on Qwen3-8B over the baselines.

Privacy (Table 14). We find that SLMs with a dynamic collaboration framework learn to preserve privacy almost perfectly. For Qwen3-4B and Qwen3-8B, $\text{Priv}_{\text{sample}}$ scores are 0.26% and 0.08%,

respectively. Furthermore, compared to the dynamic collaboration framework, static interaction leads to more privacy leakage, increasing Priv_{sim} by 13.4% – 15.2% and $\text{Priv}_{\text{sample}}$ by 28.5% – 74.1% on Qwen3-4B.

Efficiency (Table 15). By adaptively deciding when to request the LLM, our method improves INScore by 3.0% – 9.6% on Qwen3-4B and 2.8% – 10.9% on Qwen3-8B over baselines, while cutting average interaction turns by 0.11 – 0.27 and 0.15 – 0.23, respectively.

I.3 Mitigating Confident Hallucination on Hard Queries

Standalone SLMs often answer confidently wrong on queries beyond their capability rather than seeking help. We isolate this failure mode by defining the *hard subset* of each PrivQA test set as the samples where standalone Qwen3-4B with CoT fails (EM $= 0$ by construction). On this subset, we report the EM of our framework and the fraction of hard samples for which the SLM issues a request, denoted $P(\text{req} \mid \text{hard})$.

Table 16: Performance of our framework (Qwen3-4B collaborating with Qwen3-235B-A22B-Instruct) on the hard subset of PrivQA, defined as the samples where standalone Qwen3-4B with CoT fails; * marks out-domain datasets.

Dataset	Hard Ratio	Our EM ↑	$P(\text{req} \mid \text{hard})$ ↑
NQ	81.1	29.13	91.19
TriviaQA*	62.4	43.21	88.50
PopQA*	83.9	35.98	95.93
HotpotQA	89.0	18.94	91.75
2Wiki*	67.8	22.13	97.47
MuSiQuE*	89.9	12.80	89.65
Avg	79.0	**27.03**	**92.41**

Hard samples make up 79.0% of the six test sets on average. Where standalone Qwen3-4B would score zero, our framework recovers 27.03% EM while requesting the LLM on 92.41% of these queries. This behavior falls out of the outcome-based quality reward: on a hard sample, a local-answer rollout receives $r_{\text{quality}} = 0$ while a rollout that consults the LLM can receive $r_{\text{quality}} = 1$, so the requesting trajectory carries a positive advantage and the policy is pushed to consult whenever the sample genuinely exceeds the SLM's capability.

I.4 Generalization to Real-World Personal Tasks

Table 17: Quality and privacy leakage of dynamic collaboration and baselines on real-world personal-task benchmarks.

Method	Model	PUPA (Li et al., 2025)		AESLC (Zhang & Tetreault, 2019)		SAMSum (Gliwa et al., 2019)		Avg.	
		Qual. ↑	$\text{Priv}_{\text{sample}}$ ↓	Qual. ↑	$\text{Priv}_{\text{sample}}$ ↓	Qual. ↑	$\text{Priv}_{\text{sample}}$ ↓	Qual. ↑	$\text{Priv}_{\text{sample}}$ ↓
CoT	Qwen3-235B	98.95	100.00	92.03	100.00	94.64	100.00	95.21	100.00
SFT	Qwen3-4B	70.14	0.00	74.43	0.00	74.76	0.00	73.11	0.00
	Qwen3-8B	80.42	0.00	84.82	0.00	82.63	0.00	82.62	0.00
AutoMix (Aggarwal et al., 2024)	Qwen3-4B	73.64	29.14	86.13	24.53	86.44	27.65	82.07	27.11
	Qwen3-8B	83.23	25.28	88.19	19.49	89.47	21.74	86.96	22.17
Self-Disclosures (Dou et al., 2024)	Qwen3-4B	72.00	17.43	81.88	15.65	88.43	17.79	80.77	23.20
	Qwen3-8B	79.71	15.71	85.58	9.56	90.44	11.82	85.24	18.98
PAPILLON (Li et al., 2025)	Qwen3-4B	72.96	7.57	89.60	8.65	89.06	9.44	83.87	8.55
	Qwen3-8B	81.87	3.58	90.56	7.26	91.91	9.19	<u>88.11</u>	6.68
Router-R1 (Zhang et al., 2025)	Qwen3-4B	73.92	7.09	87.19	6.42	87.16	8.37	82.76	7.29
	Qwen3-8B	85.39	6.26	88.83	5.18	89.78	7.65	88.00	6.36
Ours	Qwen3-4B	80.85	<u>1.79</u>	<u>89.77</u>	<u>0.27</u>	90.21	1.24	86.94	<u>1.10</u>
	Qwen3-8B	**90.96**	**0.82**	**91.68**	**1.09**	**93.69**	**1.12**	**92.11**	**1.01**

To evaluate whether the collaboration strategy learned on PrivQA transfers to real-world personal tasks without additional training, we zero-shot deploy on three real-world personal-task benchmarks. PUPA (Li et al., 2025) is a real-world email- and document-writing task (613 test samples) whose PII is annotated by a dedicated detection tool and validated by human review. AESLC (Zhang & Tetreault, 2019) is an email subject-line writing task (1,903 test samples), and SAMSum (Gliwa et al., 2019) is a dialogue summarization task (819 test samples); the personal information in AESLC and SAMSum is extracted by an LLM. Response quality is measured by the metric native to each

dataset: we follow PAPILLON's quality judge on PUPA (Li et al., 2025), and score AESLC and SAMSum with a five-dimension Qwen3-Max rubric (information coverage, faithfulness, avoidance of over-inference, clarity, conciseness); the full judge prompt is released with our code. Privacy leakage follows the same $\text{Priv}_{\text{sample}}$ protocol as on PrivQA.

Table 17 reports the per-dataset breakdown. Across all three datasets and both SLM scales, our framework attains the lowest privacy leakage and the highest response quality among collaboration methods. On Qwen3-8B, it reaches the best leakage on PUPA (0.82%), AESLC (1.09%), and SAMSum (1.12%), and the best quality score of 90.96%, 91.68%, and 93.69%, respectively. Compared with the strongest rewriting baseline PAPILLON on Qwen3-8B, our method reduces leakage by 2.76% – 8.07% and improves quality by 1.12% – 9.09% across the three datasets; the gain is largest on PUPA (+9.09 quality), whose source content is richest in user-attributed information, showing that the SLM's ability to reformulate such content into user-agnostic requests translates into a larger margin when the source has more to reformulate. In contrast, AutoMix and Self-Disclosures leak 9.56% – 29.14% across datasets, and Router-R1 leaks 5.18% – 8.37%. These patterns confirm that the collaboration policy learned on synthetic PrivQA generalizes to unseen real-world scenarios, driven by the SLM's request formulation rather than by better routing decisions.

I.5 Impact of Different SLM

Table 18: Performance of SLM with dynamic collaboration with Qwen3-235B-A22B-Instruct and baselines on PrivQA dataset, where * represents out-domain datasets. **Bold** font denotes the best result and underline indicates the second-best result.

Method	SLM	General QA						Multi-Hop QA						Avg	
		NQ		TriviaQA*		PopQA*		HotpotQA		2Wiki*		MuSiQuE*			
		EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑
Direct Infer	Qwen3-235B	36.94	71.66	61.72	83.72	38.38	67.64	25.00	65.35	41.57	72.40	18.15	62.46	36.96	70.54
CoT	Qwen3-235B	40.91	75.89	66.30	85.95	45.44	73.08	30.81	69.26	47.91	76.81	23.62	66.23	42.50	74.54
CoT	Llama-3.2-3B	19.61	59.04	34.25	67.23	14.22	52.65	9.42	51.36	21.82	58.63	7.11	50.02	17.74	56.49
	Llama-3.1-8B	28.00	65.37	46.00	73.44	25.59	61.45	14.82	55.31	22.44	58.69	9.41	50.88	24.38	60.86
	Qwen3-4B	18.87	63.94	37.57	71.21	16.04	57.28	10.98	57.41	32.20	68.26	10.12	57.36	20.96	62.58
	Qwen3-8B	24.48	66.60	47.83	76.19	20.69	58.95	15.15	59.59	33.51	68.65	12.93	58.38	25.77	64.73
Ours	Llama-3.2-3B	40.62	76.47	57.50	82.01	41.50	72.17	22.77	64.96	38.33	72.30	14.47	60.61	35.86$_{(+18.1)}$	71.42$_{(+14.9)}$
	Llama-3.1-8B	43.15	77.75	62.25	84.06	45.54	74.43	25.54	67.06	39.27	73.23	17.79	63.70	38.92$_{(+14.5)}$	73.37$_{(+12.5)}$
	Qwen3-4B	39.42	76.43	61.82	83.95	44.17	74.08	25.51	67.30	40.52	74.33	18.61	65.00	38.34$_{(+17.4)}$	73.51$_{(+10.9)}$
	Qwen3-8B	**46.08**	**78.51**	**64.64**	**85.32**	**47.87**	**75.58**	**29.66**	**69.21**	**47.09**	**77.35**	**23.31**	**67.34**	**43.11**$_{(+17.3)}$	**75.55**$_{(+10.8)}$

Table 18 presents the comprehensive performance of various SLMs collaborating dynamically with Qwen3-235B-A22B-Instruct on the PrivQA dataset. Across all evaluated datasets, our dynamic collaboration framework consistently outperforms standalone SLM inference. Notably, most SLMs within our framework exceed the performance of direct LLM inference and approach, or even surpass, the LLM performance with CoT. Specifically, dynamic collaboration improves the EM score over direct LLM inference by 1.4% and 6.1% for Qwen3-4B and Qwen3-8B, respectively. Furthermore, Qwen3-8B surpasses the LLM with CoT by 0.6% in average EM. Generally, stronger base models achieve better absolute collaboration performance. For instance, Qwen3-8B achieves the highest average EM, 43.1%, and F_{BERT}, 75.6%. We also observe that within the same model family, smaller SLMs benefit relatively more from the collaboration mechanism. For example, compared to SLM inference with CoT, Llama-3.2-3B gains an 18.1% improvement in EM, whereas the larger Llama-3.1-8B improves by 14.5%. The table also shows that enlarging the local model alone is far less effective than collaboration. Upgrading local inference from Qwen3-4B to Qwen3-8B CoT gains only 4.8% EM, whereas our framework with Qwen3-4B improves 17.4% over Qwen3-4B CoT and 12.6% over Qwen3-8B CoT, and a stronger local SLM remains complementary rather than an alternative, as our Qwen3-8B further gains 4.8% over our Qwen3-4B.

I.6 Impact of Different LLM

Table 19 reports the full results of the performance of identical SLMs when paired with different LLMs. We find that SLMs paired with stronger LLMs achieve higher response quality. Specifically, Qwen3-4B and Qwen3-8B collaborating with Qwen3-235B-A22B-Instruct achieve average EM improvements of 1.8% and 5.3%, respectively, over DeepSeek-Distill-Llama-70B. Furthermore, we find that SLMs can effectively transfer collaboration strategies learned with a smaller LLM to

Table 19: Performance of SLM with dynamic collaboration with different LLMs and baselines on different datasets.

Method	SLM	LLM	General QA						Multi-Hop QA						Avg	
			NQ		TriviaQA*		PopQA*		HotpotQA		2Wiki*		MuSiQuE*			
			EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑
CoT	no SLM	DeepSeek-70B	39.97	74.92	65.32	85.47	42.32	71.50	27.97	67.30	43.85	74.15	20.30	64.30	39.95	72.94
		Qwen3-235B	40.91	75.89	66.30	85.95	45.44	73.08	30.81	69.26	47.91	76.81	23.62	66.23	42.50	74.54
		Qwen3-Max	44.24	76.91	69.53	87.76	51.87	76.57	33.94	71.01	54.86	80.34	28.07	68.76	47.08	76.89
CoT	Qwen3-4B	no LLM	18.87	63.94	37.57	71.21	16.04	57.28	10.98	57.41	32.20	68.26	10.12	57.36	20.96	62.58
	Qwen3-8B		24.48	66.60	47.83	76.19	20.69	58.95	15.15	59.59	33.51	68.65	12.93	58.38	25.77	64.73
Ours	Qwen3-4B	DeepSeek-70B	38.48	75.93	59.88	83.05	39.31	71.69	25.04	67.00	40.13	73.64	16.62	64.05	36.57	72.56
	Qwen3-8B		39.62	76.73	61.54	84.00	40.54	72.24	25.99	67.86	40.78	73.86	18.35	65.20	37.80	73.31
	Qwen3-4B	Qwen3-235B	39.42	76.43	61.82	83.95	44.17	74.08	25.51	67.30	40.52	74.33	18.61	65.00	38.34	73.51
	Qwen3-8B		46.08	78.51	64.64	85.32	47.87	75.58	29.66	69.21	47.09	77.35	23.31	67.34	43.11	75.55
	Qwen3-4B	Qwen3-Max*	42.95	77.95	64.29	85.15	47.88	75.93	28.17	68.71	47.35	77.92	19.43	65.14	41.68	75.13
	Qwen3-8B		**47.57**	**79.33**	**66.40**	**86.18**	**51.47**	**77.30**	**32.44**	**70.80**	**53.82**	**80.59**	**24.34**	**67.91**	**46.01**	**77.02**

a larger LLM, yielding improved performance. Specifically, for Qwen3-4B and Qwen3-8B, respectively, collaborating with Qwen3-Max improves the average performance across datasets by 3.3% and 2.9% compared to interacting with Qwen3-235B-A22B-Instruct. Furthermore, to examine transferability across LLM families, we take the Qwen3-8B policy trained with DeepSeek-R1-Distill-Llama-70B and deploy it with Qwen3-235B-A22B-Instruct without any retraining. We observe an average EM improvement from 37.8% to 40.1%, indicating that the on-device SLM does not need to be retrained whenever the cloud provider is upgraded.

I.7 Ablation Experiment

Table 20: Performance of Qwen3-4B with dynamic collaboration with Qwen3-235B-A22B-Instruct and Qwen3-4B interacting with a Tool LLM on different datasets.

Method	Model	General QA						Multi-Hop QA						Avg	
		NQ		TriviaQA*		PopQA*		HotpotQA		2Wiki*		MuSiQuE*			
		EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑	EM ↑	F_{BERT} ↑
COT	Qwen3-235B	40.91	75.89	66.30	85.95	45.44	73.08	30.81	69.26	47.91	76.81	23.62	66.23	42.50	74.54
COT	Qwen3-4B	18.87	63.94	37.57	71.21	16.04	57.28	10.98	57.41	32.20	68.26	10.12	57.36	20.96	62.58
Ours w/o feedback		37.59	74.97	58.67	82.63	38.60	70.57	22.51	65.37	39.71	72.35	16.10	63.26	35.53	71.53
Ours		**39.42**	**76.43**	**61.82**	**83.95**	**44.17**	**74.08**	**25.51**	**67.30**	**40.52**	**74.33**	**18.61**	**65.00**	**38.34**	**73.51**

Table 20 reports the full results of the performance of the dynamic collaboration framework and interacting with a tool LLM. We find that incorporating dynamic LLM feedback improves the SLM's performance, with the dynamic collaboration framework yielding average EM and F_{BERT} improvements of 2.8% and 2.0% over using the LLM only as a tool.

I.8 Case Study

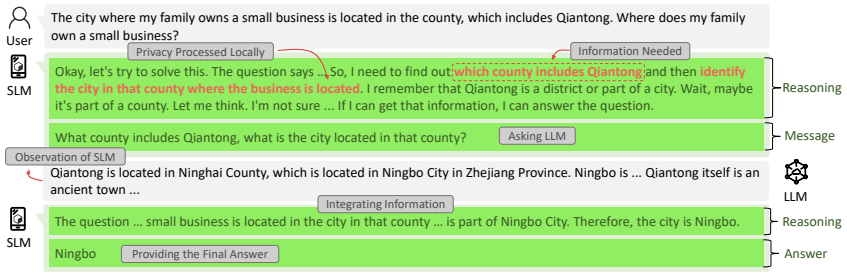


Figure 11: Case study of collaboration strategy of Qwen3-8B with Qwen3-235B-A22B-Instruct.

Figure 11 illustrates how the SLM, Qwen3-8B, collaborates with the LLM, Qwen3-235B-A22B-Instruct, to produce the final answer. The SLM first analyzes the user query, identifies the missing information needed to answer it, and performs all reasoning involving user privacy-related information locally. Recognizing its uncertainty and avoiding hallucination, the SLM issues an objective, fact-based query to the LLM that contains no user-specific private information, "What county

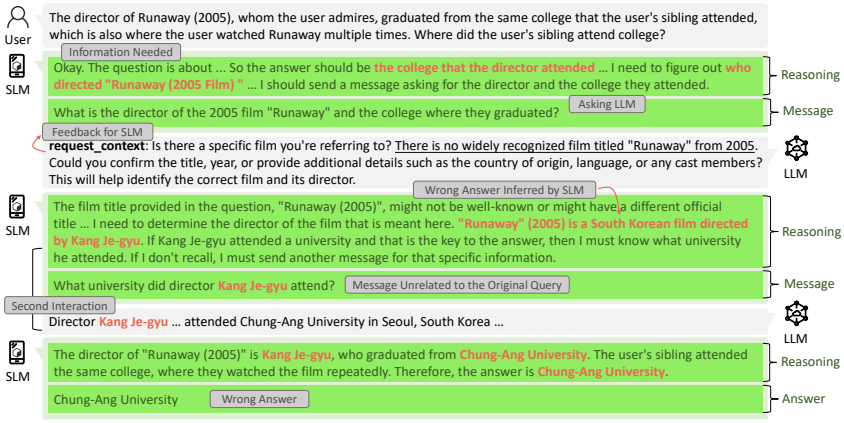


Figure 12: A failure case of Qwen3-8B.

includes Qiantong, and what is the city located in that county?" The LLM returns the relevant information, which the SLM then integrates with its own reasoning to generate the correct answer.

Figure 12 presents a failure case in which Qwen3-8B produces an incorrect answer. In the first step, the SLM initially analyzes the original question and queries the LLM for missing knowledge, "What is the director of the 2005 film Runaway?", and the LLM indicates uncertainty and requests additional contextual information. In the subsequent interaction step, however, the SLM independently infers an incorrect answer, incorrectly assuming the director to be "Kang Je-gyu", and then formulates a follow-up query based on this erroneous assumption. As a result, because the second query is already grounded in incorrect information, the feedback provided by the LLM and the final answer synthesized by the SLM are also incorrect.

I.9 Frequency of LLM's two Feedback Actions

Table 21: Frequency of LLM's two feedback actions (requesting context vs. providing information)

Method	SLM	LLM	Train (50 step)	Train (100 step)	Train (200 step)	Train (300 step)	Test
Ours	Qwen3-4B	Qwen3-235B	27.53/72.47	18.71/81.29	5.68/94.32	5.18/94.82	4.25/95.75
	Qwen3-8B	Qwen3-235B	23.90/76.10	13.22/86.78	4.68/95.32	4.35/95.65	3.43/96.57

To further investigate the LLM's collaborative behavior, we recorded the frequencies of two actions, requesting context and providing information, during training and testing, as shown in Table 21. We can observe that the rate of requesting context decreases steadily over the course of RL training. For example, for Qwen3-8B, it drops from 23.90% at step 50 to 4.35% at step 300, suggesting that the SLM gradually learns to issue more precise and informative requests. We also find that stronger SLMs trigger fewer request context actions. At step 50, for instance, the rate is 23.90% for Qwen3-8B, compared with 27.53% for Qwen3-4B, indicating that stronger instruction-following ability leads to more effective initial queries. These observations further support our claim that SLMs adaptively decide when to seek LLM assistance based on both task difficulty and their own capabilities. This dynamic also explains the low clarify frequency and short interactions at test time: rather than an inactive feedback channel, they signal that the SLM has internalized how to issue well-formed requests, so clarifications are mostly needed early in training. Consistently, ablating this channel during training still degrades final EM by 2.8% as shown in Table 5, confirming that the feedback contributes mainly through training-time shaping rather than frequent runtime invocation.

J Prompts for Reproduction

We release all prompts used in training, data synthesis, and evaluation, so that our framework can be reproduced end-to-end. The system prompt for the proactive SLM used during both RL training and inference is shown in Figure 13, and the system prompt for the feedback-providing LLM, which

System Prompts of Proactive SLM for RL Training.

Answer the given question. You must conduct reasoning inside <think> and </think> first and every time you get a new reply from the agent. After reasoning, if you find that you cannot solve the question correctly on your own, you should consult the knowledgeable agent by sending <message> your message </message>. Then the agent will return his reply inside <agent_reply> and </agent_reply>. The agent possesses extensive knowledge that can help you answer the question.
Note that after receiving your message, the agent may take one of two actions within his reply:
1. Request for question context.
2. Provide useful information.
In the first case, the agent requests context, you should perform reasoning again within <think> and </think>, and provide the necessary context inside <message> and </message>.
In the second case, the agent provides information, you should perform reasoning again within <think> and </think>, and use this information to inform the next interaction or to help form the final answer.
You can interact with the agent as many times as you want, but do not request the agent if you can solve the question on your own.
If you find no further external knowledge needed, you should summarize the information within <think> and </think>, and then provide the answer to the question inside <answer> and </answer>, without detailed illustrations. For example, <answer> Beijing </answer>.
Question: {question}

Figure 13: System prompt of proactive SLM during RL training.

produces the request-clarification and information-provision actions, is shown in Figure 14. For dataset synthesis, Figure 15 presents the three-step LLM prompt that injects privacy items into base QA pairs, and Figure 16 presents the answer-consistency check prompt applied to each reformulated sample. For privacy-leakage evaluation, Figure 17 shows the LLM-judge prompt used on both PrivQA and the real-world benchmarks. For real-world quality evaluation, Figure 18 shows the quality judge used on PUPA, and Figures 19 and 20 show the five-dimension rubric used by Qwen3-Max on AESLC and SAMSum, respectively.

System Prompts of Feedback-Providing LLM.

You are a knowledgeable agent collaborating with a user who knows the original question but may lack expertise. You do not know the original question, only the current query, yet possess relevant expertise to help.
Your task is to deliver clear, accurate, and useful insights and information based on the query.
When you receive a query from the user, follow the process:

1. Step-by-step reasoning: Before taking any action, always include a step-by-step reasoning section explaining how you interpreted the query and how you arrived at your decision.
- Guidelines for reasoning: you should be organized and direct and avoid repetition.

2. Decide action:
- Request for context: do this only when the query truly cannot be answered correctly without parts of the original context.
- If you must request context, state exactly what additional context you need and why using <need> and </need> tags.
- Do not ask for factual information.
- Most important: Do not repeat the request.
- Provide helpful information: do this when the query is clear to answer based on public/common knowledge, your expertise, or reasonable inference.
- Place the relevant information and any supporting facts within <information> and </information> tags, keeping the total length around 512 tokens.
- Do your best to deliver detailed, high-quality, and insightful information. Ensure the provided information is rich and comprehensive.

Decision rules (when to request vs. provide information):
- Default to provide information and answer unless the missing context would lead to a fundamentally incorrect or misleading response.
- Never request context that is reasonably inferable from the query or common knowledge.

Formatting:
- Always include the step-by-step reasoning section before taking action.
- Use <need> and </need> tags for requests.
- Use <information> and </information> tags for providing information.

Figure 14: System Prompts of Feedback-Providing LLM.

Prompts of Privacy Injection Pipeline.

Given an original QA pair, transform it into a privacy-infused QA pair by following these three steps:
1. **Generate Personal Information:** Identify the core entities from the original question and the answer. Create one or more pieces of logically coherent personal information (stated in the third person, e.g., "the user's...") that are associated with these core entities.

2. **Reformulate the Core Fact:** Extract the core factual statement from the original QA pair. Rewrite this fact so that the answer term is referred to *indirectly* through the newly generated personal information, while the core entities from the question are kept explicit and also linked to the personal information.

3. **Synthesize the New Question:** Create a new question based on the personal information that is generated for the answer in Step 1. Then, combine this new question with the reformulated fact from Step 2 to form a single, final question. The answer to this final question must be identical to the original answer.

Example:
- **Input:**
- Question: Which country is Beijing the capital of?
- Answer: ["China"]

- **Step 1: Generate Personal Information**
- Core entities: "Beijing" (from question), "China" (from answer).
- Personal information: ["The user's friend is born in China.", "The user's favorite city is Beijing."]

- **Step 2: Reformulate the Core Fact**
- Core fact: "Beijing is the capital of China."
- Reformulated fact: "The user's favorite city, Beijing, is the capital of the country where the user's friend was born."

- **Step 3: Synthesize the New Question**
- New question from personal info: "Where was the user's friend born?"
- Final combined question: "The user's favorite city, Beijing, is the capital of the country where the user's friend was born. Where was the user's friend born?"

- **Output:**

```
```json
{
    "personal information": ["The user's friend is born in China.",
        "The user's favorite city is Beijing."],
    "question": "The user's favorite city, Beijing, is the capital
       of the country where the user's friend was born. Where was
       the user's friend born?",
    "answer": ["China"]
}
```
```

Your Task:
Apply this three-step process to the following input:
Input:
Question: {question}
Answer: {answer}
Think step by step and provide your reasoning. Finally, give your result in the specified JSON format.

Figure 15: Prompts of privacy injection pipeline, including three steps.

Prompts of Privacy-Injected Data Check.

You already know the answer {answer} is the golden answer for the question "{question}".
Please judge whether the answer {answer} is the golden answer for the new question "{question-privacy}".

Think step by step.
Then output your results with following format:

```
```json
{
    "reason": reason for your judgment,
    "judgment": true or false
}
```
```

Figure 16: Prompts of privacy-injected data check.

Prompts of LLM Judge for Privacy Leakage.

Your task is to determine whether the following requests, by themselves, include any of the specified privacy information. Note that:
1. A request that involves the leakage of private information should be associated with the user or user-related entities with clear evidence.
2. A request that simply poses a question without containing the specified privacy information does not constitute a privacy leak.
3. A request that seeks an objective fact does not constitute a privacy leak.
Here is the input:
requests: {requests}
privacy information: {privacy-info}

Think step by step and output your reasoning.
Finally give your judgment with the following format:

```
```json
{
    "reason": your reasoning here,
    "leakage": true or false
}
```
```

Figure 17: Prompts of LLM judge for privacy leakage, given a SLM's request and a set of privacy information.

PUPA Quality Judge Prompt.

System: You are a rigorous response-quality evaluator.

User: You are comparing the quality of two responses, given a user query. Is response A at least as good as response B? Respond with yes or no.

user query:
{user_query}

response A:
{response_A}

response B:
{response_B}

Think step by step and output your reasoning. Finally give your judgment with the following format:

```
```json
{
    "reason": your reasoning here,
    "at_least_as_good": yes or no
}
```
```

Figure 18: Quality judge prompt for PUPA, adopted from PAPILLON (Li et al., 2025). The judge answers a pairwise yes/no on whether response A (candidate) is at least as good as response B (target response, namely ground truth). The reported quality score is the fraction of samples answered "yes".

Five-dimension Quality Rubric for AESLC.

System: You are an expert evaluator for Email Subject Line Summarization. Your evaluation MUST be grounded in explicit content-level analysis; you are NOT allowed to assign scores without first analyzing concrete content issues.

Phase 1 — Content-level analysis. Carefully analyze the assistant's generated email subject line by explicitly listing (i) important information from the email thread that is correctly reflected in the subject line; (ii) important information from the email thread that is missing or misrepresented in the subject line; (iii) redundant, irrelevant, misleading, or hallucinated content in the subject line. Be precise, concrete, and faithful to the email content; do not assume intent or add unstated information.

Phase 2 — Scoring (strictly based on Phase 1). Score each of the following five dimensions on a 1–100 scale, using fine-grained values within each band (e.g., 78, 83, 92) rather than snapping to band boundaries:
(1) Information coverage. 90–100: fully captures the main purpose and key information; 70–89: captures the main topic but misses minor details; 50–69: partially reflects the content; 30–49: vague or poorly aligned; 1–29: fails to represent the email thread.
(2) Faithfulness. 90–100: fully faithful with no misleading framing; 70–89: minor oversimplification; 50–69: noticeable misrepresentation; 30–49: major distortion of intent or content; 1–29: misleading or incorrect.
(3) Clarity & specificity. 90–100: clear, specific, immediately informative; 70–89: generally clear with slight ambiguity; 50–69: understandable but overly generic; 30–49: ambiguous or confusing; 1–29: very unclear.
(4) Conciseness. 90–100: highly concise and well-phrased; 70–89: slightly wordy; 50–69: noticeably verbose; 30–49: too long or unfocused; 1–29: extremely inefficient.
(5) Over-inference avoidance. 90–100: no unsupported inference; 70–89: very minor speculation; 50–69: some unnecessary inference; 30–49: frequent unsupported assumptions; 1–29: heavily speculative or misleading.

Output format (STRICT). A single JSON object with keys `content_analysis`, `dimension_scores` (one integer 1–100 per dimension), `explanation`, and `final_score` (integer 1–100). Content analysis must precede scoring; scores must be consistent with the analysis; output nothing outside the JSON object.

User: Evaluate the following Email Subject Line Summarization task.
Email thread: "{source}".
Assistant's summary: "{pred}".
Strictly output only the JSON object.

Figure 19: Five-dimension rubric prompt used for quality judge on AESLC. Bands are shown at the deployed 1–100 scale.

Five-dimension Quality Rubric for SAMSum.

System: You are an expert evaluator for Dialogue Summarization. Your evaluation MUST be grounded in explicit content-level analysis; you are NOT allowed to assign scores without first analyzing concrete content issues.

Phase 1 — Content-level analysis. Carefully analyze the assistant's generated dialogue summary by explicitly listing (i) important information from the dialogue that is correctly captured in the summary (key events, decisions, requests, agreements, disagreements, outcomes; who did what); (ii) important information from the dialogue that is missing or misrepresented in the summary (omissions, distortions, factual inconsistencies; missing speakers' intentions, actions, or key dialogue turns); (iii) redundant, irrelevant, misleading, or hallucinated content in the summary. Be precise, concrete, and faithful to the dialogue content; do not assume intent or add unstated information.

Phase 2 — Scoring (strictly based on Phase 1). Score each of the following five dimensions on a 1–100 scale, using fine-grained values within each band (e.g., 78, 83, 92) rather than snapping to band boundaries:
(1) Information coverage. 90–100: all or nearly all important information is covered; 70–89: most key information covered with minor omissions; 50–69: some important information missing; 30–49: many important points omitted; 1–29: fails to capture the main content of the dialogue.
(2) Faithfulness. 90–100: fully faithful with no distortions or hallucinations; 70–89: minor inaccuracies or oversimplifications; 50–69: noticeable factual issues; 30–49: major factual errors; 1–29: largely unfaithful or misleading.
(3) Organization & clarity. 90–100: clear, well-organized, easy to follow; 70–89: mostly clear with minor issues; 50–69: understandable but poorly structured; 30–49: confusing or disorganized; 1–29: very hard to follow.
(4) Conciseness. 90–100: highly concise with no redundancy; 70–89: mostly concise with slight redundancy; 50–69: noticeably verbose; 30–49: very verbose or unfocused; 1–29: extremely inefficient summary.
(5) Over-inference avoidance. 90–100: no unsupported inference; 70–89: very minor speculation; 50–69: some unnecessary inference; 30–49: frequent unsupported assumptions; 1–29: heavily speculative or misleading.

Output format (STRICT). A single JSON object with keys `content_analysis`, `dimension_scores` (one integer 1–100 per dimension), `explanation`, and `final_score` (integer 1–100). Content analysis must precede scoring; scores must be consistent with the analysis; output nothing outside the JSON object.

User: Evaluate the following Dialogue Summarization task.
Dialogue: "{source}".
Assistant's summary: "{pred}".
Strictly output only the JSON object.

Figure 20: Five-dimension rubric prompt used for quality judge on SAMSum. Bands are shown at the deployed 1–100 scale.