

CoSteer: Collaborative Decoding-Time Personalization via Local Delta Steering

Hang Lv^{1*}, Sheng Liang^{2*}, Hao Wang^{1†}, Hongchao Gu¹, Yaxiong Wu²,
Wei Guo², Defu Lian¹, Yong Liu², Enhong Chen^{1†}

¹University of Science and Technology of China

²Huawei Technologies Co., Ltd.

Abstract

Personalization has become crucial for adapting models to the diverse and evolving needs of users across cultural, temporal, and contextual dimensions. While existing methods often rely on centralized fine-tuning or static preference alignment within a single model, they struggle to achieve both real-time and high-quality personalization under the resource and privacy constraints of personal devices. To address this challenge, we propose **CoSteer**, a collaborative framework that enables tuning-free, real-time personalization via decoding-time adaptation. By leveraging logit differences between context-aware and -agnostic local small models, CoSteer steers cloud-based large models, ensuring effective personalization while preserving the large model’s capabilities. Personalization is handled locally, with only final tokens sent to the cloud, maintaining both user context and system efficiency. Through extensive experiments across a wide range of tasks, we demonstrate that CoSteer generates high-quality personalized content, ensuring both effectiveness and computational efficiency. Our results highlight its robustness across models and environments, confirming its practical applicability in real-world scenarios.

1 Introduction

Personalization enables AI systems to move beyond generic content by tailoring interactions to individual users. Personalized generation creates customized content from user-specific attributes, including linguistic patterns, interaction histories, and contextual preferences (Xu et al., 2025). It supports recommender systems, adaptive dialogue, and content creation platforms (Lyu et al., 2023; Zhang et al., 2024a; Wu et al., 2025; Mysore et al., 2024) in diverse real-world settings.

*Equal contribution. Code and data are available at <https://github.com/USTC-StarTeam/CoSteer>.

†Correspondence: wanghao3@ustc.edu.cn.

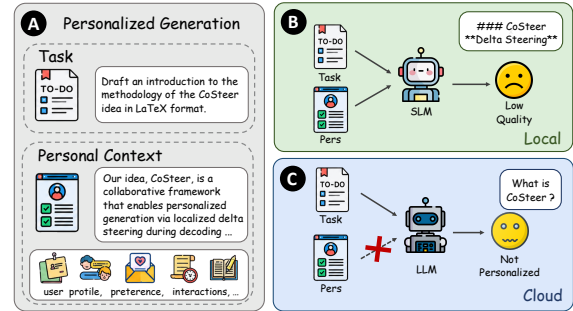


Figure 1: The dilemma of edge-cloud personalization. Local SLMs protect privacy but yield low-quality outputs, whereas cloud LLMs offer high quality but lack access to private context.

When users issue queries, the request can be handled either by a local model on the device or by a cloud-hosted large model. As illustrated in Figure 1, local models can access private user context, but their small size and limited capacity results in suboptimal generation quality (Niu et al., 2025; Chen et al., 2026). Cloud-based models produce high-quality text but cannot incorporate sensitive local context, yielding outputs that are not personalized (Zhao et al., 2025; Guo et al., 2026).

Prior work has focused on two complementary aspects. One line investigates how cloud-based models can incorporate user-specific information through retrieval-augmented generation (Richardson et al., 2023; Wang et al., 2023a) or preference alignment (Zhou et al., 2024; Wu et al., 2023), enhancing personalization while exposing sensitive user data. Another line explores strategies to improve the generation quality of small on-device models via lightweight parameter updates or adapters (Zhang et al., 2025a; Zhong et al., 2021), yet these methods are limited by computational resources and training data, making it challenging to adapt to users’ evolving preferences in real time. Both approaches fail to balance personalization quality and privacy protection, hindering their deployment in sensitive real-world scenarios.

To address these fundamental challenges, we propose **CoSteer**, a collaborative framework for real-time personalized generation via **localized delta steering**. Our core innovation leverages the logit difference between personal context-aware and -agnostic outputs from on-device small language models (SLMs) as dynamic steering signals for cloud-based LLMs. Specifically, the SLM produces two predictions: 1) personalized outputs incorporating private user context, and 2) generic outputs using only the query. The resulting logit differentials guide the cloud model toward personalized outputs. CoSteer integrates these signals during decoding as an online, token-level refinement, enabling edge devices to iteratively adjust cloud predictions without transmitting raw personal context or intermediate representations.

Extensive evaluations across **five model pairs** and **eight datasets** validate CoSteer’s broad applicability. Our results show that even a tiny local model can effectively steer a massive cloud LLM, achieving remarkable “weak-to-strong” generalization and outperforming conventional baselines. Beyond raw performance gains, CoSteer demonstrates practical advantages for deployment: it enables collaboration across heterogeneous model architectures, maintains robustness against noisy contexts, and significantly reduces computational overhead compared to full-context cloud processing.

We summarize our key contributions as follows:

- **The CoSteer Framework:** We propose CoSteer, a novel and tuning-free framework that reformulates decoding-time adaptation as an online learning process. By utilizing lightweight local logit deltas to iteratively steer cloud model predictions, it enables precise personalization while maintaining strict data isolation.
- **Extensive Experimental Validation:** We evaluate CoSteer across 5 model pairs and 8 datasets, showing strong performance over conventional baselines and demonstrating that even when the local model is over $50\times$ smaller than the cloud LLM, it can effectively steer generation.
- **Practical Deployment and Generalization:** We rigorously validate CoSteer’s robustness to noisy user contexts, its adaptability across heterogeneous architectures, and computational efficiency alongside two streamlined variants, confirming its reliable and practical deployability in diverse real-world scenarios.

2 Related work

Training-time Personalization. Mainstream personalization methods typically rely on training-time adaptation, using user-specific data to tailor model behavior. One approach applies shared PEFT modules across users (Zhang et al., 2025a; Woźniak et al., 2024; Zhu et al., 2024; Kong et al., 2025), while per-user PEFTs tune isolated parameters for individualized adaptation, achieving better performance (Zhong et al., 2021; Peng et al., 2024; Tan et al., 2025, 2024). Beyond PEFT, reinforcement learning (RL) aligns models with user preferences through reward-driven optimization. Recent work casts personalization as a multi-objective RL problem: methods like MORLHF (Wu et al., 2023) and MODPO (Zhou et al., 2024), including dynamic policy merging frameworks (Jang et al., 2023; Ramé et al., 2023; Shi et al., 2024; Chen et al., 2025). These methods can combine multiple objectives or policies to better capture diverse user preferences. While effective, these approaches are computationally intensive, rely on static datasets, and recent studies show that even strong models struggle with long-horizon preference following and context-sensitive personalization (Guo et al., 2026; Yoon et al., 2026; Garbacea et al., 2026).

Inference-time Adaptation. Inference-time adaptation modulates model behavior during inference to bypass retraining. Traditional approaches often augment the input prompt with retrieved user profiles or historical interactions (Richardson et al., 2023; Wang et al., 2023a), while logit-steering methods adjust token distributions on-the-fly using differential signals computed on-device. Examples include Linear Alignment (Gao et al., 2024) and Contrastive Decoding (Li et al., 2023), which preserve privacy locally, CoS (He et al., 2025), which scales personal information with an adaptable parameter, and Amulet (Zhang et al., 2025b), which iteratively optimizes logits to align with user preferences. EXACT (Yu et al., 2026) retrieves explicit user attributes to guide generation, CLIPer (Su et al., 2026) uses external classifiers to shape outputs according to user preferences, and Preference Heads (Zhang et al., 2026) masks internal preference heads for interpretable personalization. Applying these methods solely on small local models yields suboptimal performance, while applying them on cloud LLMs requires transmitting user attributes or context, introducing privacy risks.

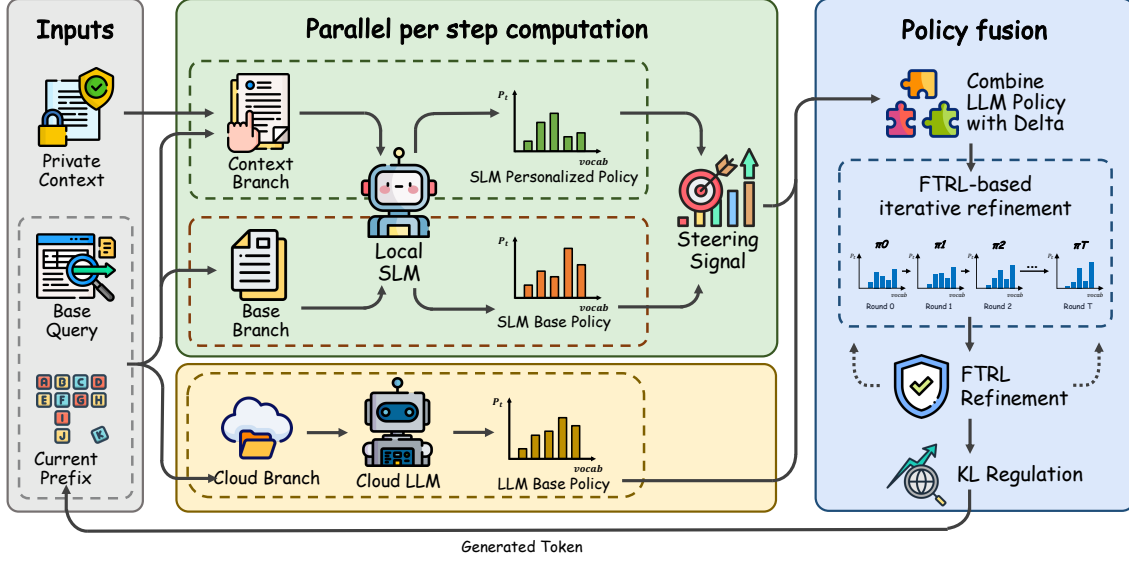


Figure 2: Overview of the CoSteer framework. At each decoding step, the local SLM generates a private delta steering signal to refine the cloud LLM’s base policy via on-device policy fusion.

Collaborative Generation. The computational constraints of edge devices have driven innovative paradigms in collaborative generation between cloud-based LLMs and local SLMs (Narayan et al., 2025; Niu et al., 2025; Zhan et al., 2025; Chen et al., 2026). Existing approaches focus on two technical routes: assistive alignment, where local SLMs augment the LLM’s user preference alignment, and reward-guided decoding, where lightweight reward models provide preference signals. Aligner (Ji et al., 2024) injects user-specific linguistic patterns via SLM feedback, while Expo (Zheng et al., 2025) aligns outputs through linear interpolation of layer-wise weights. Recent advancements further demonstrate that training specialized small reward models can effectively steer LLM outputs toward personalized objectives (Snell et al., 2024; Liu et al., 2025). Among these, Proxy-tuning (Liu et al., 2024) and Cogensis (Zhang et al., 2024b) are closest to our design. Proxy-tuning modifies the LLM’s logits distribution by contrasting pre-trained and fine-tuned SLM outputs, whereas Cogensis employs a learned fusion network to combine logits from both models. In contrast to these existing methods which require tuning or additional modules to align local and cloud models, our proposed CoSteer framework establishes a completely tuning-free collaboration mechanism, thereby thoroughly eliminating the need for any additional training.

To clearly illustrate the technical positioning of CoSteer, we provide Table 3 in Appendix A.1, where our distinctions and contributions are discussed in more detail.

3 Methodology

3.1 Preliminary

3.1.1 Task formulation

We consider personalized text generation under a strict edge–cloud deployment: a cloud-hosted LLM only processes the raw user query p_{base} , while a locally deployed SLM has access to privacy-sensitive personal context p_{pers} (e.g., user profiles and interaction histories) that must remain on-device. Our goal is to improve generation quality by combining the LLM’s general capability with localized personalization, without sending p_{pers} to the cloud.

To enable test-time adaptation under the privacy boundary, we model decoding as a sequence of per-token online optimization problems. At each decoding step, the model chooses the next token $a \in \mathcal{A}$ according to a policy $\pi(\cdot | p, s)$, where the policy is parameterized by the LLM logit distribution. The per-token objective is:

$$\pi^* = \arg \max_{\pi \in \Pi} \mathbb{E}_{a \sim \pi(\cdot | p, s)} [r(a | p, s)], \quad (1)$$

where p denotes a generic prompt, s is the already generated prefix, and r is an (unknown) reward function reflecting the current user’s personal context and preferences.

3.1.2 Online learning

Personalization signals are individual-specific and may change over time, making purely offline training on static data insufficient for test-time adaptation. Online learning provides a natural abstraction: at each step, the algorithm updates the policy using

newly observed utility signals, aiming to minimize regret against the best fixed policy in hindsight.

Inspired by Amulet (Zhang et al., 2025b), we adopt Follow-The-Regularized-Leader (FTRL) (McMahan, 2011) to refine the token policy during decoding. In its generic form, FTRL updates the policy at iteration t by optimizing the cumulative utilities with an explicit regularizer:

$$\pi_t = \arg \max_{\pi \in \Pi} \left[\sum_{i=0}^{t-1} \mathcal{U}_i(\pi) - \frac{1}{\eta} \Omega(\pi) \right], \quad (2)$$

where $\mathcal{U}_i$ is the utility at update iteration i , $\Omega(\cdot)$ is a regularizer, and η controls the update strength. In our setting, the true reward r in Eq. 1 is not directly observable at inference time, so $\mathcal{U}_i$ must be constructed from an efficient proxy signal derived from available information. Throughout the paper, t denotes the current inner FTRL update iteration within a single decoding step, while i indexes the past inner iterations in the cumulative sum; the decoding token position is omitted for simplicity.

3.2 Decoding-time personalization via local delta steering

To instantiate the online optimization at decoding time, we need a utility function that can be evaluated under a strict edge–cloud privacy boundary: the cloud-hosted LLM only conditions on the raw query p_{base} , while the privacy-sensitive personal context p_{pers} must remain on-device. We therefore design a logit-level steering signal that is computed locally and can be used to refine the token distribution without exposing p_{pers} .

A series of recent works, including Contrastive Decoding (Li et al., 2023) and Linear Alignment (Gao et al., 2024), show that contrasting logits with and without a certain context provides an effective direction for improving generation (He et al., 2025; Zhang et al., 2025b). From our online learning view, such a contrast can be written as the following utility:

$$u_t(a) = \alpha (\log \pi_t(a) - \log \pi_{\text{base}}(a)). \quad (3)$$

However, directly computing this contrast on the cloud LLM would require transmitting p_{pers} to the cloud, which violates our privacy constraint. Our key observation is that the cloud LLM and the on-device SLM share the same vocabulary and decode under the same prefix s , so a context-induced delta computed by the SLM can be used as a compatible steering signal for the LLM logit space. We thus define a dual-contrastive utility as follows:

$$u_t(a) = \underbrace{\alpha (\log \pi_t(a) - \log \pi_{\text{base}}(a))}_{\text{LLM Policy Contrast}} + \underbrace{\beta (\log \pi_{\text{pers}}^*(a) - \log \pi_{\text{base}}^*(a))}_{\text{SLM Delta Steering}}. \quad (4)$$

Here, p_{base} is the user query and p_{pers} is the private personal context. We formalize three policies: (1) the target policy $\pi_t(a)$ to be optimized; (2) the LLM base policy $\pi_{\text{base}}(a) = P_{\text{LLM}}(a \mid p_{\text{base}}, s)$; and (3) the SLM reference policies $\pi_{\text{base}}^*(a) = P_{\text{SLM}}(a \mid p_{\text{base}}, s)$ and $\pi_{\text{pers}}^*(a) = P_{\text{SLM}}(a \mid p_{\text{base}}, p_{\text{pers}}, s)$. The first term encourages controlled deviation from the LLM base behavior, while the second term injects personalization through the on-device delta $\log \pi_{\text{pers}}^*(a) - \log \pi_{\text{base}}^*(a)$.

To retain the LLM’s general capability and prevent overly aggressive steering, we add a KL constraint to the objective:

$$\mathcal{U}_t(\pi) = u_t(\pi) - \lambda D_{\text{KL}}(\pi \parallel \pi_0), \quad (5)$$

where we set the reference policy as $\pi_0 = \pi_{\text{base}}$. Using KL divergence also naturally matches the regularization form required by FTRL, resulting in a proximal-style update:

$$\pi_t = \arg \max_{\pi \in \Pi} \left[\sum_{i=0}^{t-1} \mathcal{U}_i(\pi) - \frac{1}{\eta} D_{\text{KL}}(\pi \parallel \pi_{t-1}) \right]. \quad (6)$$

Although Eq. 6 suggests an iterative refinement, performing expensive inner-loop optimization during inference hinders interactive personalization. We therefore derive a closed-form solution, yielding an efficient logit-level fusion rule:

$$\pi_t(a) \propto \exp \left(\frac{1}{t\lambda + \frac{1}{\eta}} \left(\sum_{i=0}^{t-1} u_i(a) + t\lambda \log \pi_0(a) + \frac{1}{\eta} \log \pi_{t-1}(a) \right) \right). \quad (7)$$

The derivation is provided in Appendix A.2. Equation 7 can be implemented efficiently as element-wise operations in log-probability space followed by a single normalization, making it suitable for on-device execution.

We refer to the resulting edge–cloud decoding procedure—where the device uses the SLM delta and Eq. 7 to select each token while the cloud LLM is fixed—as **CoSteer**. The concrete inference pipeline is described in Section 3.3.

3.3 CoSteer inference pipeline

We now present the comprehensive **CoSteer** framework by detailing its edge-cloud instantiation (illustrated in Figure 2 and Algorithm 1) based on the closed-form fusion rule derived in Eq. 7. To maintain strict data isolation, the cloud-hosted LLM receives only the raw query p_{base} and computes the base policy π_{base} . Concurrently, the edge device, which securely holds the personal context p_{pers} , runs the local SLM twice to generate the personalized policy π_{pers}^* and the base policy π_{base}^* . The log-probability difference between these SLM outputs forms the local delta steering signal (Eq. 4). Upon receiving the cloud logits, the device applies the closed-form update (Eq. 7) to compute the target policy π_t . The next token is then sampled and transmitted back to the cloud to continue decoding.

This architecture enforces a rigid privacy boundary. Because policy fusion occurs entirely on-device, the cloud never observes p_{pers} , the raw SLM distributions, or the local delta. CoSteer naturally accommodates optional local safeguards (e.g., PII filters) prior to token transmission, achieving decoding-time personalization while keeping both the cloud LLM and privacy boundaries intact.

Algorithm 1 CoSteer Framework

Require:

Cloud-based LLM policy generator π_{LLM} , local SLM policy generator π_{SLM} , user query p_{base} , personal context p_{pers} , current sequence s , max tokens M , hyperparameters: $T, \alpha, \beta, \lambda, \eta > 0$

Ensure: Personalized sequence s

```

1: Initialize  $s \leftarrow \emptyset$ 
2: repeat
3:    $\pi_{base} \leftarrow \pi_{LLM}(a|p_{base}, s)$  ▷ Cloud computation
4:    $\pi_{base}^* \leftarrow \pi_{SLM}(a|p_{base}, s)$  ▷ Edge computation
5:    $\pi_{pers}^* \leftarrow \pi_{SLM}(a|p_{base}, p_{pers}, s)$ 
6:    $\Delta \leftarrow \log \pi_{pers}^* - \log \pi_{base}^*$ 
7:    $\pi_0 \leftarrow \pi_{base}$  ▷ Initialize policy
8:   for  $t = 1$  to  $T$  do
9:      $u_t \leftarrow \alpha(\log \pi_{t-1} - \log \pi_{base}) + \beta\Delta$ 
10:    Update policy using Equation 7
11:   end for
12:    $\pi^* \leftarrow \pi_T$  ▷ Final optimized policy
13:   Sample  $a \sim \pi^*$ , Update  $s \leftarrow s \circ a$ 
14: until  $\text{len}(s) \geq M$  or EOS generated
15: return  $s$ 

```

4 Experiment

4.1 Tasks and datasets

To demonstrate the versatility of our CoSteer framework, we conduct extensive experiments on **eight** datasets spanning two major categories of personalization tasks: personalized content generation and preference alignment.

Personalized content generation We utilize the official test sets of two benchmarks. Cogensis (Zhang et al., 2024b) guides personalized generation using summarized profiles and past experiences. LongLaMP (Kumar et al., 2024) focuses on personalized long-form generation across three tasks: abstract (Tang et al., 2008), review (Ni et al., 2019), and topic writing (Völske et al., 2017). We augment each LongLaMP query with the top-5 historical records retrieved via bge-reranker-v2-m3 (Chen et al., 2024) as its personal context.

Preference alignment We evaluate on four datasets: HelpSteer (Wang et al., 2023b), Truthful QA (Lin et al., 2022), UltraChat (Ding et al., 2023), and Personal Preference Eval (Gao et al., 2024). These datasets require generating content aligned with explicitly stated user preferences. Due to the large scale of these datasets, we randomly sample 200 test instances from each dataset.

Detailed dataset description and examples are shown in Appendix B.5

4.2 Evaluation metrics

To ensure fairness in comparison, we employ task-specific metrics proposed by these datasets themselves for evaluation. For Cogensis, we use GPT-4o to evaluate the response’s overall and personalized scores, averaging the results over five runs with temperature set to 0 to mitigate potential instability. For LongLaMP, we employ ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005) scores to measure content overlap and semantic alignment, supplementing them with human evaluation (detailed in Appendix B.6) to account for the potential limitations of these automatic metrics. For preference alignment tasks, following prior work (Zhang et al., 2025b; Zhong et al., 2024), we set user preferences to be *concise*, *creative*, *uplifting*, and *verbose*, and utilize the reward model ArmoRM-8B (Wang et al., 2024) to evaluate the degree to which the generations align with these preferences.

Models	Setting	Cogenesis		Abstract Generation			Review Writing			Topic Writing			Pref Align			
		Ovl	Per	R-1	R-L	MET	R-1	R-L	MET	R-1	R-L	MET	creative	verbose	concise	uplifting
Qwen 7B-1.5B	SLM w/o	6.63	6.21	36.48	17.74	27.57	20.40	10.39	10.39	25.21	11.09	17.51	.5441	.5154	.6104	.5632
	SLM w/	7.81	7.63	39.75	22.03	27.76	23.08	12.40	12.94	22.89	11.46	17.12	.6214	.6998	.7102	.6594
	LLM w/o	8.00	7.63	39.81	20.53	25.56	30.15	14.04	17.71	27.64	11.93	21.49	.7058	.6931	.7039	.7183
	CoSteer	8.44	8.50	42.98	23.61	28.20	32.72	15.92	20.36	25.93	12.38	22.84	.7915	.7608	.7404	.7885
	LLM w/	8.62	8.60	44.50	24.63	31.15	33.83	15.55	22.42	28.82	13.77	23.44	.8884	.8076	.7693	.8414
Qwen 32B-7B	SLM w/o	8.00	7.63	39.81	20.53	25.56	30.15	14.04	17.71	27.64	11.93	21.49	.7058	.6931	.7039	.7183
	SLM w/	8.62	8.60	44.50	24.63	31.15	33.83	15.55	22.42	28.82	13.77	23.44	.8884	.8076	.7693	.8414
	LLM w/o	8.12	7.87	40.66	21.02	26.61	32.21	14.44	19.61	28.82	12.20	21.16	.7020	.6873	.7225	.7146
	CoSteer	8.78	8.64	45.41	26.04	33.52	34.88	15.89	26.51	30.10	14.52	24.20	.8589	.8532	.7274	.8579
	LLM w/	8.83	8.76	43.33	23.47	30.10	34.65	15.74	22.77	30.73	14.20	24.25	.9017	.8193	.7912	.8538
Llama 8B-1B	SLM w/o	7.04	6.55	33.20	18.20	28.55	31.75	14.92	19.78	20.81	10.21	17.30	.6535	.6355	.5900	.6646
	SLM w/	7.69	7.52	39.81	21.53	30.11	32.36	15.02	22.06	20.17	10.58	18.64	.7981	.7908	.7037	.8065
	LLM w/o	7.69	7.13	39.33	20.69	29.41	34.58	15.32	22.10	26.93	12.35	21.82	.7038	.6878	.6765	.7116
	CoSteer	7.29	7.73	41.28	24.97	31.19	31.68	13.68	24.57	26.11	12.00	23.69	.8911	.8582	.7812	.8721
	LLM w/	8.61	8.44	43.91	23.93	32.01	36.39	15.95	23.56	30.54	14.02	23.81	.8869	.8298	.7909	.8551
Qwen 8B-0.6B	SLM w/o	6.84	6.64	37.13	20.41	21.00	24.41	12.83	12.39	24.19	11.88	15.74	.5710	.5359	.5981	.5937
	SLM w/	7.85	7.79	42.48	23.25	26.58	25.04	13.36	13.02	26.17	13.46	18.40	.6386	.6432	.6000	.6436
	LLM w/o	8.27	7.99	40.76	21.23	25.77	30.89	14.47	16.98	28.47	12.85	21.49	.7597	.7530	.6929	.7611
	CoSteer	8.62	8.65	41.11	22.36	25.86	32.24	14.84	18.47	29.03	13.61	22.98	.7619	.7573	.6913	.7648
	LLM w/	8.96	8.96	43.99	23.69	29.48	35.42	15.71	21.59	31.48	15.34	24.78	.8895	.8031	.7713	.8552
Qwen 32B-0.6B	SLM w/o	6.84	6.64	37.13	20.41	21.00	24.41	12.83	12.39	24.19	11.88	15.74	.5710	.5359	.5981	.5937
	SLM w/	7.85	7.79	42.48	23.25	26.58	25.04	13.36	13.02	26.17	13.46	18.40	.6386	.6432	.6000	.6436
	LLM w/o	8.31	8.22	41.05	21.88	26.23	30.23	14.08	16.41	29.01	12.51	21.44	.7861	.7850	.6934	.7783
	CoSteer	8.52	8.56	43.74	23.69	30.97	30.89	14.14	18.87	26.75	13.17	21.32	.7858	.7883	.6873	.7790
	LLM w/	9.09	9.03	44.46	23.95	30.15	35.12	15.60	20.65	32.63	15.37	25.07	.9095	.8095	.8040	.8654

Table 1: Comparative performance across eight personalized content generation and preference alignment tasks. Metrics include overall (Ovl) and personalized (Per) scores for Cogenesis, ROUGE-1/-L (R-1/-L) and METEOR (MET) for Longlamp datasets, and averaged alignment scores for four user preferences. **Bold** entries indicate that CoSteer outperforms the three baseline methods. Gray values represent the privacy-violating near-upper-bound performance, yet underlined CoSteer values surpasses these incompatible references.

4.3 Settings and baselines

To verify the robustness of our framework, we evaluate five model pairs of varying scales and architectures, including (1) Qwen2.5-7B-Instruct with Qwen2.5-1.5B-Instruct (2) Qwen2.5-32B-Instruct with Qwen2.5-7B-Instruct (Qwen et al., 2025) (3) Llama3.1-8B-Instruct with Llama3.2-1B-Instruct (Grattafiori et al., 2024) (4) Qwen3-8B with Qwen3-0.6B and (5) Qwen3-32B with Qwen3-0.6B (Yang et al., 2025).

We establish the following critical baselines per pair: (1) SLM w/o: Base performance of standalone SLMs without personal context. (2) SLM w/: SLMs augmented with personal context. (3) LLM w/o: Cloud-based LLMs without access to personal context. (4) LLM w/: The near-upper-bound performance where LLMs directly access personal context. It is important to note that this approach can lead to privacy breaches and does not align with our task setting.

4.4 Main Results

Table 1 summarizes the average performance across all five model-pair configurations. Detailed per-dataset scores (Table 5) and the full statistical significance analysis (Table 6) are deferred to Appendices B.2 and B.3, respectively.

Consistent and substantial gains. Across the vast majority of settings, CoSteer significantly outperforms two key baseline families: (1) cloud-based LLMs responding without personal context, and (2) local SLMs with or without context. This indicates that CoSteer can reliably translate locally stored user context into measurable improvements in personalized generation and preference alignment.

Robustness across standard model pairs. On common pairings (e.g., Qwen2.5 32B-7B and 7B-1.5B), the improvements are especially consistent and robust, suggesting that CoSteer is insensitive to a specific architecture or capacity range, and generalizes well under typical deployment scales.

	Abstract			Review			Writing		
	R-1	R-L	MET	R-1	R-L	MET	R-1	R-L	MET
LA/CS	41.49	25.57	29.96	26.60	13.68	17.59	24.09	<u>12.32</u>	19.94
SFT	40.71	21.85	27.00	<u>31.55</u>	12.91	<u>20.23</u>	25.53	10.80	19.35
Proxy tuning	40.76	21.37	25.95	29.54	<u>14.16</u>	16.79	27.13	12.11	<u>21.38</u>
CoSteer	42.98	<u>23.61</u>	<u>28.20</u>	32.72	15.92	20.36	<u>25.93</u>	12.38	22.84
w/o alpha	41.10	21.78	26.09	30.51	14.40	17.47	27.41	12.32	21.95
w/o KL	42.52	22.76	27.04	31.44	14.43	18.70	26.47	13.12	22.82

Table 2: Performance comparison of our Costeer framework against other methods on LongLamp benchmark, with an ablation study on its key components. Best results are marked in **bold** while the second are underlined.

Effective steering under extreme capacity gaps.

Even with the ultra-compact Qwen3-0.6B, CoSteer yields clear benefits on complex personalized writing tasks. This demonstrates that our collaborative framework successfully leverages the strong reasoning and generation capabilities of the large base LLM, while effectively injecting essential personal context from the tiny SLM.

Approaching the near-upper-bound. Remarkably, CoSteer often approaches or even exceeds the “near-upper-bound” achieved by LLMs with full context access. These results demonstrate that CoSteer enables cloud-based LLMs to generate personalized content using only locally stored user context. We further analyze the interactions among task complexity and model scale in Appendix B.4.

4.5 Comparison with Alternative Methods

While CoSteer is **unique in its technical positioning** as discussed in Appendix A.1, to comprehensively demonstrate its superiority, we further compare it against other alignment methods, particularly those that rely solely on localized SLMs. Among the various existing techniques, we report on a selection of methods that can be applied to Personalized Content Generation tasks and that outperform SLM w/ baseline. These methods include:

(1) **Linear Alignment (Gao et al., 2024) and Context Steering (He et al., 2025)**: These two methods are formally equivalent to scaling the logit difference of the SLM (with and without personal context) and applying it to its own inference process. This constitutes an inference-time optimization performed directly on the SLM;

(2) **Supervised Fine-Tuning (SFT)**: This involves directly fine-tuning the localized SLMs using personal data. Although we argue that fine-tuning a separate model for each user and task is impractical in real-world scenarios due to privacy and data constraints, we report its performance here for a thorough academic comparison;

(3) **Proxy-tuning (Liu et al., 2024)**: This method leverages the difference in the SLM’s logit distributions before and after the SFT process described above. This difference is added to the LLM’s logits to steer its output distribution.

To comprehensively evaluate CoSteer against these alternative methods, we conduct experiments across all five LLM–SLM configurations, with the complete results compiled in Appendix B.8 (Table 8). Due to space constraints, we limit our presentation in the main text to the first configuration (Qwen7B–1.5B), as shown in Table 2.

Quantitative results confirm CoSteer’s consistent superiority across diverse model pairs. This robustness demonstrates its ability to seamlessly blend the privacy-preserving strengths of local SLMs with the advanced capabilities of cloud LLMs in a completely tuning-free manner.

4.6 Ablation Study and Parameter Analysis

Ablation studies We conduct an ablation studies to validate the contributions of the key components in our framework. The SLM delta steering signal (the β term in Eq. (4)) is the central mechanism of CoSteer. Our study therefore focuses on the impact of individually removing two other components: the LLM Alignment Term (the α term) and the KL Divergence Regularizer from the FTRL algorithm. The results in Table 2 proved their effectiveness. A broader ablation on the entire iterative FTRL (we name it LightCosteer) is discussed in Section 5.3.

Hyperparameters We examine hyperparameter impacts via abstract generation using the same Qwen7B–1.5B configuration and the ROUGE-L metric. When varying a single parameter, all others remain at default. Results in Figure 3 and Appendix B.9 show consistently strong and stable performance across broad parameter ranges, confirming that our default settings (Appendix B.1) are highly robust and well-calibrated.

5 Practical Deployment Considerations

In Section 4, we showed the empirical superiority of the CoSteer framework through extensive experiments and comparisons. In this section, we focus on its practical viability, analyzing its feasibility and performance from various angles that simulate real-world conditions and constraints.

5.1 Robustness to Noisy User Context

In practical on-device applications, the user-specific context is often imperfect and susceptible to noise. To evaluate CoSteer’s resilience in such unpredictable scenarios, we apply two distinct noise conditions (detailed in Appendix B.10). Against **realistic noise**, simulated via a weaker BM25 retriever, CoSteer consistently maintains its performance edge over baselines (Table 9). Furthermore, against extreme **adversarial noise** (injecting completely irrelevant context), the framework degrades gracefully rather than collapsing (Table 10). This confirms that the strong LLM guidance prevents faulty local contexts from completely misleading the token generation, demonstrating a robustness vital for real-world deployment.

5.2 Generalization across Model Scales and Architectures

CoSteer generalizes well across diverse model configurations, which is essential for real-world deployment. We validate this in two dimensions.

Generalization Across Scales As shown in Table 1, pairing a powerful 32B LLM with a highly compact 0.6B SLM (Qwen3-0.6B) yields strong performance across all testbeds. Despite being over $50\times$ smaller, the SLM effectively steers the massive LLM, confirming the framework’s remarkable robustness to extreme capacity gaps.

Collaboration Across Architectures To accommodate scenarios where LLMs and SLMs possess divergent tokenizers, we investigate two alignment strategies (Appendix B.11). The first, CoSteer_map, aligns models via vocabulary intersection. Because its applicability is limited when tokenizers are highly divergent, we propose CoSteer_byte, which operates on a shared, vocabulary-agnostic byte representation (Hayase et al., 2025). Table 11 shows both strategies successfully enable cross-architecture guidance, confirming that CoSteer is not confined to homogeneous model families.

5.3 Efficiency and Practical Variants

A practical edge-cloud framework must strictly manage computational and communication overheads. By processing the private context exclusively on the local SLM, CoSteer avoids appending long user histories to the massive remote LLM. This architectural choice substantially reduces the LLM’s attention workload. As estimated in Table 14, for a 1000-token context, the privacy-violating LLM w/ baseline incurs approximately $3.3\times$ more FLOPs (Narayanan et al., 2021) than CoSteer, demonstrating significant computational savings in deployment.

To optimize latency, CoSteer employs an asynchronous two-stream pipeline (Shown in Figure 4) where the local SLM inference is largely masked by the cloud network round-trip. Nevertheless, the iterative FTRL updates ($T = 20$) and per-token communication can still introduce bottlenecks. To address these constraints, we propose two streamlined variants (fully analyzed in Appendix B.12):

(1) **LightCoSteer**: Designed for edge devices with severe compute limits, this variant eliminates the iterative optimization by setting $T = 1$. It simplifies the framework to a single-step logit adjustment, providing a lightweight yet effective directional steering signal.

(2) **AdaCoSteer**: Designed to minimize communication overhead, this variant adaptively deactivates steering mid-generation. It leverages the observation that once the LLM’s token confidence consistently exceeds a high threshold (Song et al., 2025), the local context is often redundant. Terminating early allows the LLM to generate subsequent standard tokens autonomously.

Together, these variants establish a flexible performance-efficiency spectrum, empowering practitioners to dynamically balance personalization quality against strict resource constraints in real-world deployments.

6 Conclusion

In this work, we propose **CoSteer**, a tuning-free collaborative framework that achieves personalized generation by steering an LLM via closed-form logit deltas from an SLM. Extensive experiments validate its effectiveness and robustness across diverse model configurations. Furthermore, by introducing streamlined variants to reduce computational and communication overheads, CoSteer provides a practical solution for deployment.

Limitations

A primary limitation of our work lies in the scope of its privacy protection. While CoSteer strictly isolates raw personal context, local parameters, and logit distributions on the edge device, a residual vulnerability exists during token transmission: the generated discrete tokens sent back to the cloud may implicitly leak personalized attributes. Addressing this risk necessitates additional safeguards in future developments, such as applying on-device PII redactors or formal token-level Differential Privacy to sanitize outputs prior to upload. CoSteer’s localized architecture leaves open the possibility of incorporating such protective measures for highly sensitive applications.

Acknowledgments

The work was supported by grants from New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0123403), National Natural Science Foundation of China (No. U24A20253) and Scientific Research Innovation Capability Support Project for Young Faculty.

This work was also supported by the National Natural Science Foundation of China (Nos. U23A20319, 62472394, and 62441227), as well as the Anhui Province Science and Technology Innovation Project (Nos. 202423k09020010 and 202423k09020011).

References

- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. [Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). *Preprint*, arXiv:2402.03216.
- Ruizhe Chen, Xiaotian Zhang, Meng Luo, Wenhao Chai, and Zuozhu Liu. 2025. [Pad: Personalized alignment of llms at decoding-time](#). *Preprint*, arXiv:2410.04070.
- Yining Chen, Jihao Zhao, Bo Tang, Haofen Wang, Yue Zhang, Fei Huang, Feiyu Xiong, and Zhiyu Li. 2026. [Memprivacy: Privacy-preserving personalized memory management for edge-cloud agents](#). *Preprint*, arXiv:2605.09530.
- Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. [Enhancing chat language models by scaling high-quality instructional conversations](#). *Preprint*, arXiv:2305.14233.
- Songyang Gao, Qiming Ge, Wei Shen, Shihan Dou, Junjie Ye, Xiao Wang, Rui Zheng, Yicheng Zou, Zhi Chen, Hang Yan, Qi Zhang, and Dahua Lin. 2024. [Linear alignment: A closed-form solution for aligning human preferences without tuning and feedback](#). *Preprint*, arXiv:2401.11458.
- Cristina Garbacea, Heran Wang, and Chenhao Tan. 2026. [Personalized benchmarking: Evaluating llms by individual preferences](#). *Preprint*, arXiv:2604.18943.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Qianyun Guo, Yibo Li, Yue Liu, and Bryan Hooi. 2026. [Towards realistic personalization: Evaluating long-horizon preference following in personalized user-llm interactions](#). *Preprint*, arXiv:2603.04191.
- Jonathan Hayase, Alisa Liu, Noah A. Smith, and Sewoong Oh. 2025. [Sampling from your language model one byte at a time](#). *Preprint*, arXiv:2506.14123.
- Jerry Zhi-Yang He, Sashrika Pandey, Mariah L. Schrum, and Anca Dragan. 2025. [Context steering: Controllable personalization at inference time](#). *Preprint*, arXiv:2405.01768.
- Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2023. [Personalized soups: Personalized large language model alignment via post-hoc parameter merging](#). *Preprint*, arXiv:2310.11564.
- Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Juntao Dai, Tianyi Qiu, and Yaodong Yang. 2024. [Aligner: Efficient alignment by learning to correct](#). *Preprint*, arXiv:2402.02416.
- Minbeom Kim, Kang-il Lee, Seongho Joo, Hwaran Lee, Thibaut Thonet, and Kyomin Jung. 2025. [Drift: Decoding-time personalized alignments with implicit user preferences](#). *ArXiv*, arXiv:2502.14289.
- Xiaoyu Kong, Jiancan Wu, An Zhang, Leheng Sheng, Hui Lin, Xiang Wang, and Xiangnan He. 2025. [Customizing language models with instance-wise lora for sequential recommendation](#). *Preprint*, arXiv:2408.10159.
- Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A. Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang,

- Shubham Agarwal, Nedim Lipka, Chien Van Nguyen, Thien Huu Nguyen, and Hamed Zamani. 2024. [Longlamp: A benchmark for personalized long-form text generation](#). *Preprint*, arXiv:2407.11016.
- Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. [Contrastive decoding: Open-ended text generation as optimization](#). *Preprint*, arXiv:2210.15097.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. [Truthfulqa: Measuring how models mimic human falsehoods](#). *Preprint*, arXiv:2109.07958.
- Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024. [Tuning language models by proxy](#). *Preprint*, arXiv:2401.08565.
- Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. 2025. [Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling](#). *Preprint*, arXiv:2502.06703.
- Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Christopher Leung, Jiajie Tang, and Jiebo Luo. 2023. [Llm-rec: Personalized recommendation via prompting large language models](#). *Preprint*, arXiv:2307.15780.
- Brendan McMahan. 2011. [Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization](#). In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 525–533, Fort Lauderdale, FL, USA. PMLR.
- Sheshera Mysore, Zhuoran Lu, Mengting Wan, Longqi Yang, Bahareh Sarrafzadeh, Steve Menezes, Tina Baghaee, Emmanuel Barajas Gonzalez, Jennifer Neville, and Tara Safavi. 2024. [Pearl: Personalizing large language model writing assistants with generation-calibrated retrievers](#). *Preprint*, arXiv:2311.09180.
- Avanika Narayan, Dan Biderman, Sabri Eyuboglu, Avner May, Scott Linderman, James Zou, and Christopher Re. 2025. [Minions: Cost-efficient collaboration between on-device and cloud language models](#). *Preprint*, arXiv:2502.15964.
- Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Anand Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. 2021. [Efficient large-scale language model training on gpu clusters using megatron-lm](#). *Preprint*, arXiv:2104.04473.
- Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. [Justifying recommendations using distantly-labeled reviews and fine-grained aspects](#). In *Conference on Empirical Methods in Natural Language Processing*.
- Chaoyue Niu, Yucheng Ding, Junhui Lu, Zhengxiang Huang, Hang Zeng, Yutong Dai, Xuezhen Tu, Chengfei Lv, Fan Wu, and Guihai Chen. 2025. [Collaborative learning of on-device small model and cloud-based large model: Advances and future directions](#). *Preprint*, arXiv:2504.15300.
- Dan Peng, Zhihui Fu, and Jun Wang. 2024. [Pocketllm: Enabling on-device fine-tuning for personalized llms](#). *Preprint*, arXiv:2407.01031.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Alexandre Ramé, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord. 2023. [Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards](#). *Preprint*, arXiv:2306.04488.
- Chris Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. 2023. [Integrating summarization and retrieval for enhanced personalization via large language models](#). *Preprint*, arXiv:2310.20081.
- Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hannaneh Hajishirzi, Noah A. Smith, and Simon S. Du. 2024. [Decoding-time language model alignment with multiple objectives](#). *Preprint*, arXiv:2406.18853.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. [Scaling llm test-time compute optimally can be more effective than scaling model parameters](#). *Preprint*, arXiv:2408.03314.
- Feifan Song, Shaohang Wei, Wen Luo, Yuxuan Fan, Tianyu Liu, Guoyin Wang, and Houfeng Wang. 2025. [Well begun is half done: Low-resource preference alignment by weak-to-strong decoding](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12654–12670, Vienna, Austria. Association for Computational Linguistics.
- Jinyan Su, Jinpeng Zhou, Claire Cardie, and Wen Sun. 2026. [Cliper: Tailoring diverse user preference via classifier-guided inference-time personalization](#). *Preprint*, arXiv:2605.07162.
- Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. 2024. [Personalized pieces: Efficient personalized large language models through collaborative efforts](#). *Preprint*, arXiv:2406.10471.

- Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. 2025. [Democratizing large language models via personalized parameter-efficient fine-tuning](#). *Preprint*, arXiv:2402.04401.
- Jie Tang, Jing Zhang, Limin Yao, Juan-Zi Li, Li Zhang, and Zhong Su. 2008. [Arnetminer: extraction and mining of academic social networks](#). In *Knowledge Discovery and Data Mining*.
- Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. [TL;dr: Mining reddit to learn automatic summarization](#). In *NFiS@EMNLP*.
- Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024. [Interpretable preferences via multi-objective reward modeling and mixture-of-experts](#). *Preprint*, arXiv:2406.12845.
- Hongru Wang, Rui Wang, Fei Mi, Yang Deng, Zezhong Wang, Bin Liang, Ruifeng Xu, and Kam-Fai Wong. 2023a. [Cue-cot: Chain-of-thought prompting for responding to in-depth dialogue questions with llms](#). *Preprint*, arXiv:2305.11792.
- Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, and Oleksii Kuchaiev. 2023b. [Help-steer: Multi-attribute helpfulness dataset for steerlm](#). *Preprint*, arXiv:2311.09528.
- Stanisław Woźniak, Bartłomiej Koptyra, Arkadiusz Janz, Przemysław Kazienko, and Jan Kocoń. 2024. [Personalized large language models](#). *Preprint*, arXiv:2402.09269.
- Bowen Wu, Wenqing Wang, Haoran Li, Ying Li, Jingsong Yu, and Baoxun Wang. 2025. [Interpersonal memory matters: A new task for proactive dialogue utilizing conversational history](#). *Preprint*, arXiv:2503.05150.
- Zequiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. [Fine-grained human feedback gives better rewards for language model training](#). *Preprint*, arXiv:2306.01693.
- Yiyan Xu, Jinghao Zhang, Alireza Salemi, Xinting Hu, Wenjie Wang, Fuli Feng, Hamed Zamani, Xiangnan He, and Tat-Seng Chua. 2025. [Personalized generation in large model era: A survey](#). *ArXiv*, arXiv:2503.02614.
- 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, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Sangyeon Yoon, Sunkyoung Kim, Hyesoo Hong, Wonje Jeung, Yongil Kim, Wooseok Seo, Heuiyeon Yeen, and Albert No. 2026. [Benchpres: A benchmark for context-aware personalized preference selectivity of persistent-memory llms](#). *Preprint*, arXiv:2603.16557.
- Xin Yu, Hanwen Xing, and Lingzhou Xue. 2026. [Exact: Explicit attribute-guided decoding-time personalization](#). *Preprint*, arXiv:2602.17695.
- Junfei Zhan, Haoxun Shen, Zheng Lin, and Tengjiao He. 2025. [Prism: Privacy-aware routing for adaptive cloud-edge llm inference via semantic sketch collaboration](#). *Preprint*, arXiv:2511.22788.
- Jizhi Zhang, Keqin Bao, Wenjie Wang, Yang Zhang, Wentao Shi, Wanhong Xu, Fuli Feng, and Tat-Seng Chua. 2024a. [Prospect personalized recommendation on large language model-based agent platform](#). *Preprint*, arXiv:2402.18240.
- Kai Zhang, Yejin Kim, and Xiaozhong Liu. 2025a. [Personalized llm response generation with parameterized memory injection](#). *Preprint*, arXiv:2404.03565.
- Kaiyan Zhang, Jianyu Wang, Ermo Hua, Biqing Qi, Ning Ding, and Bowen Zhou. 2024b. [Cogenesis: A framework collaborating large and small language models for secure context-aware instruction following](#). *Preprint*, arXiv:2403.03129.
- Weixu Zhang, Ye Yuan, Changjiang Han, Yuxing Tian, Zipeng Sun, Linfeng Du, Jikun Kang, Hong Kang, Xue Liu, and Haolun Wu. 2026. [Preference heads in large language models: A mechanistic framework for interpretable personalization](#). *Preprint*, arXiv:2604.22345.
- Zhaowei Zhang, Fengshuo Bai, Qizhi Chen, Chengdong Ma, Mingzhi Wang, Haoran Sun, Zilong Zheng, and Yaodong Yang. 2025b. [Amulet: Realignment during test time for personalized preference adaptation of LLMs](#). In *The Thirteenth International Conference on Learning Representations (ICLR)*.
- Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. 2025. [Do llms recognize your preferences? evaluating personalized preference following in llms](#). *Preprint*, arXiv:2502.09597.
- Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. 2025. [Model extrapolation expedites alignment](#). *Preprint*, arXiv:2404.16792.
- Wanjuan Zhong, Duyu Tang, Jiahai Wang, Jian Yin, and Nan Duan. 2021. [UserAdapter: Few-shot user learning in sentiment analysis](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1484–1488, Online. Association for Computational Linguistics.
- Yifan Zhong, Chengdong Ma, Xiaoyuan Zhang, Ziran Yang, Haojun Chen, Qingfu Zhang, Siyuan Qi, and Yaodong Yang. 2024. [Panacea: Pareto alignment via preference adaptation for llms](#). *Preprint*, arXiv:2402.02030.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. 2024. [Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization](#). *Preprint*, arXiv:2310.03708.

Jiachen Zhu, Jianghao Lin, Xinyi Dai, Bo Chen, Rong Shan, Jieming Zhu, Ruiming Tang, Yong Yu, and Weinan Zhang. 2024. [Lifelong personalized low-rank adaptation of large language models for recommendation](#). *Preprint*, arXiv:2408.03533.

A Technical Discussion and Proofs

Method	Training Free?	Weak-to-Strong Collaborative?	No Additional Modules?
Linear Alignment (Gao et al., 2024)	Yes	No	Yes
Contrastive Decoding (Li et al., 2023)			
Context Steering (He et al., 2025)			
Amulet (Zhang et al., 2025b)	No ^a	No	No
PAD (Chen et al., 2025)			
Drift (Kim et al., 2025)			
Proxy-tuning (Liu et al., 2024)	No ^b	Yes	Yes
Cogenesis (Zhang et al., 2024b)	No ^c	Yes	No
CoSteer (Ours)	Yes	Yes	Yes

Table 3: Our Technical position. The main body of the table uses “Yes/No” for clarity. Explanations for specific “No” entries are as follows: ^aRequires a trained reward model. ^bRequires a fine-tuned smaller language model (SLM). ^cRequires a trained fusion network.

A.1 Technical Positioning and Contributions

As summarized in Table 3, **CoSteer** occupies a unique position in the landscape of personalized generation by simultaneously satisfying three critical requirements for edge-cloud deployment: it is weak-to-strong collaborative, completely tuning-free, and requires no additional modules.

Clarification of Technical Positioning Our framework distinguishes itself from existing literature along two primary axes:

- **Beyond Single-Model Steering:** While inference-time steering methods (e.g., Context Steering (He et al., 2025), Linear Alignment (Gao et al., 2024), Amulet (Zhang et al., 2025b)) are tuning-free, they operate strictly within a single-model paradigm where a model steers itself. CoSteer introduces a fundamental shift toward **heterogeneous cross-model collaboration**, demonstrating that a tiny, local SLM can effectively steer a massive, remote LLM to resolve the privacy-utility conflict.
- **Beyond Training-Dependent Collaboration:** Existing collaborative paradigms (e.g., Cogenesis (Zhang et al., 2024b), PAD (Chen et al., 2025), Proxy-tuning (Liu et al., 2024)) achieve cross-model interaction but heavily rely on trained fusion networks, reward models, or supervised fine-tuning (SFT) of the proxy models. CoSteer eliminates these constraints. It is **completely tuning-free**, avoiding the prohibitive computational overhead and data requirements of per-user or per-task training at the edge.

Multi-Faceted Technical Contributions To realize this unique positioning, our work makes specific, novel technical contributions that extend beyond the standard application of optimization solvers:

- **Formulation for Distributed Coordination:** Rather than claiming the FTRL optimizer (McMahan, 2011) as our invention, our core algorithmic contribution lies in repurposing this online learning framework for distributed, privacy-preserving coordination. We introduce a novel utility function (Equation 4) specifically tailored to the edge-cloud constraint. By leveraging local logit deltas as dynamic steering signals, we bridge isolated models and coordinate their generation processes without sharing parameters or transmitting raw personal data.
- **Scenario-Specific System Innovations:** To ensure this collaborative architecture is practically viable in real-world edge environments, we introduce critical system-level solutions to address unique deployment bottlenecks. We present **Ada-CoSteer**, which utilizes token confidence for adaptive early termination to significantly reduce communication latency. Furthermore, we develop **Byte-Level Fusion**, a vocabulary-agnostic strategy that technically resolves tokenizer mismatches, thereby enabling seamless collaboration across entirely divergent model architectures (e.g., Llama and Qwen).

A.2 Mathematical Derivations

Similar to Zhang et al. (2025b), We try to solve the closed-form solution of Equation 6:

$$\begin{aligned} \mathcal{L}(\pi_t, \mu) = & \underbrace{\sum_{i=0}^{t-1} \sum_{a \in A} \pi_t(a) u_i(a)}_{(1)} \\ & - t\lambda \underbrace{\sum_{a \in A} \pi_t(a) \log \frac{\pi_t(a)}{\pi_0(a)}}_{(2)} \\ & - \frac{1}{\eta} \underbrace{\sum_{a \in A} \pi_t(a) \log \frac{\pi_t(a)}{\pi_{t-1}(a)}}_{(3)} \\ & + \mu \underbrace{\left(1 - \sum_{a \in A} \pi_i(a)\right)}_{(4)}. \end{aligned} \quad (8)$$

Here, (1) and (2) originate from the utility function $\mathcal{U}$, (3) from the KL divergence, and (4) constrains the sum to 1, with the Lagrange multiplier μ . We calculate the derivation of the function for a given a , we have

$$\begin{aligned} \frac{\partial \mathcal{L}(\pi_t, \mu)}{\partial \pi_t(a)} = & \sum_{i=0}^{t-1} u_i(a) - t\lambda \left(\log \frac{\pi_t(a)}{\pi_0(a)} + 1 \right) \\ & - \frac{1}{\eta} \left(\log \frac{\pi_t(a)}{\pi_{t-1}(a)} + 1 \right) - \mu. \end{aligned} \quad (9)$$

Rearrange the terms:

$$\begin{aligned} & \sum_{i=0}^{t-1} u_i(a) - t\lambda \log \pi_t(a) \\ & + t\lambda \log \pi_0(a) - \frac{1}{\eta} \log \pi_t(a) \\ & + \frac{1}{\eta} \log \pi_{t-1}(a) - t\lambda - \frac{1}{\eta} - \mu = 0. \end{aligned} \quad (10)$$

Combine the coefficients of $\log \pi_t(a)$:

$$\begin{aligned} & - \left(t\lambda + \frac{1}{\eta} \right) \log \pi_t(a) \\ = & - \sum_{i=0}^{t-1} u_i(a) \\ & - t\lambda \log \pi_0(a) - \frac{1}{\eta} \log \pi_{t-1}(a) \\ & + t\lambda + \frac{1}{\eta} + \mu. \end{aligned} \quad (11)$$

Solve for $\log \pi_t(a)$:

$$\begin{aligned} \log \pi_t(a) = & \frac{1}{t\lambda + \frac{1}{\eta}} \left(\sum_{i=0}^{t-1} u_i(a) + t\lambda \log \pi_0(a) \right. \\ & \left. + \frac{1}{\eta} \log \pi_{t-1}(a) - t\lambda - \frac{1}{\eta} - \mu \right). \end{aligned} \quad (12)$$

let

$$C = -\frac{t\lambda + \frac{1}{\eta} + \mu}{t\lambda + \frac{1}{\eta}} \quad (13)$$

we have

$$\begin{aligned} \log \pi_t(a) = & \frac{1}{t\lambda + \frac{1}{\eta}} \left(\sum_{i=0}^{t-1} u_i(a) + t\lambda \log \pi_0(a) \right. \\ & \left. + \frac{1}{\eta} \log \pi_{t-1}(a) \right) + C. \end{aligned} \quad (14)$$

Thus

$$\begin{aligned} \pi_t(a) \propto \exp \left(\frac{1}{t\lambda + \frac{1}{\eta}} \left(\sum_{i=0}^{t-1} u_i(a) + t\lambda \log \pi_0(a) \right. \right. \\ \left. \left. + \frac{1}{\eta} \log \pi_{t-1}(a) \right) \right). \end{aligned} \quad (15)$$

Specifically, let $T = 1$ and $\beta^* = \frac{\beta}{\lambda + \frac{1}{\eta}}$, the policy is simplified to

$$\begin{aligned} \pi(a) \propto \exp \left(\log \pi_0(a) \right. \\ \left. + \beta^* (\log \pi_{pers}^*(a) - \log \pi_{base}^*(a)) \right) \end{aligned} \quad (16)$$

Parameter	Value
T	20
α	2.0
β	1.0
λ	2
η	10

Table 4: Default hyperparameter of CoSteer

B Experiment Details

B.1 Hyperparameters and Settings

To ensure the results are easily reproducible, we set `do_sample=False` for all methods. For the Qwen3 series, we set `enable_thinking=False`. Hyperparameters of CoSteer are listed in Table 4

B.2 Detailed Results

Detailed results on HelpSteer, Personal_Eval, Truthful QA and Ultrachat are shown in Table 5.

B.3 Statistical Significance Testing

To formally validate the robustness of our findings, we performed paired t-tests comparing **CoSteer** against each of the three key baselines: LLM w/o, SLM w/, and SLM w/o. The tests were conducted on the full set of evaluation samples, comparing the performance scores on a per-sample basis to determine if the observed improvements were due to more than random chance. We used a standard significance level of $\alpha = 0.05$. The results confirmed that the performance gains of **CoSteer** over all three baselines are statistically significant. In Table 6, asterisks (*) denote these significant improvements.

B.4 Detailed Analysis on Experiment Results

Analysis on Task Complexity Our framework demonstrates distinct performance patterns across task complexity levels. For context-intensive generation tasks (Cogenesis and LongLaMP) requiring deep integration of extended user profiles, histories, and multiple writing examples, larger model pairs demonstrate superior contextual reasoning capabilities. Qwen 32B-7B shows significant gains across three datasets compared to compact models (Qwen 7B-1.5B/ Llama 8B-1B). In contrast, the Llama8B-1B configuration occasionally underperforms baselines by failing to distill essential personalization signals from sparse contexts. Conversely, in preference alignment tasks where personal context reduces to concise and explicit textual instructions, compact models perform comparably to larger counterparts. This stems from the larger

models’ over-alignment when processing simplistic personalization signals and effectively overfitting to sparse preference indicators.

Analysis on Model Sizes Our approach has demonstrated excellent collaborative results across models of various sizes. An interesting finding is that the benefits of integrating personal information vary with different model sizes. In the Abstract Generation and Review Writing datasets, the performance gain of incorporating personal context with the Qwen-2.5-7B-Instruct model was significantly greater than that with the 32B model. In such cases, by using our CoSteer method to influence the token distribution of the 32B model through the logits delta derived from the 7B model, the final performance significantly outperforms 32B model with full context, yielding a relative improvement of nearly 14 percentage points in METEOR scores across these datasets.

B.5 Dataset Description

To ensure a comprehensive evaluation relevant to Cloud-LLM serving scenarios, we selected datasets covering a broad spectrum of personal AI agent applications:

- **Daily Assistance (Cogenesis):** This benchmark simulates daily tasks such as drafting emails and notifications based on user profiles and history.
- **Complex Content Creation (LongLaMP):** This focuses on reasoning-intensive, long-form generation tasks including academic abstracts, product reviews, and blog posts. These tasks typically require the superior generation capabilities of Cloud LLMs to ensure high quality.
- **General Preference Alignment:** We use four datasets (e.g., HelpSteer, TruthfulQA) to evaluate general instruction following based on specific user constraints.

We present a real example from each dataset to help readers understand the task of each dataset. Cogenesis is shown in Figure 5. Three datasets of LongLaMP are in Figure 6. Four datasets of the preference alignment task are in Figure 7.

B.6 Human Evaluation on LongLaMP

To provide a more robust validation of our method, we conducted a small-scale human evaluation. Due

Models	Setting	HelpSteer				Personal_Eval				Truthful QA				Ultrachat			
		creative	verbose	concise	uplifting	creative	verbose	concise	uplifting	creative	verbose	concise	uplifting	creative	verbose	concise	uplifting
Qwen 7B-1.5B	SLM w/o	.5609	.5301	.6437	.5638	.6043	.5882	.6556	.6722	.4468	.4231	.5349	.4544	.5642	.5202	.6074	.5625
	SLM w/	.6309	.7093	.7110	.6568	.6564	.7402	.7686	.7449	.5419	.6515	.6358	.5758	.6562	.6982	.7252	.6602
	LLM w/o	.7368	.7223	.7228	.7338	.7550	.7462	.7005	.8124	.5739	.5752	.6744	.5782	.7576	.7285	.7179	.7489
	CoSteer	.8093	.7787	.7731	.8055	.8472	.7945	.7525	.8739	.6834	.6930	.6837	.6682	.8259	.7770	.7522	.8064
	LLM w/	.8918	.8173	.7853	.8516	.9217	.8396	.8001	.9073	.8363	.7593	.7069	.7630	.9037	.8140	.7850	.8438
Qwen 32B-7B	SLM w/o	.7368	.7223	.7228	.7338	.7550	.7462	.7005	.8124	.5739	.5752	.6744	.5782	.7576	.7285	.7179	.7489
	SLM w/	.8918	.8173	.7853	.8516	.9217	.8396	.8001	.9073	.8363	.7593	.7069	.7630	.9037	.8140	.7850	.8438
	LLM w/o	.7141	.7003	.7406	.7129	.7548	.7461	.7129	.8136	.5877	.5844	.6993	.5925	.7512	.7183	.7373	.7393
	CoSteer	.8630	.8516	.7478	.8684	.9158	.8754	.7700	.9166	.7985	.8425	.6575	.7818	.8581	.8433	.7344	.8646
	LLM w/	.9038	.8270	.8104	.8628	.9254	.8434	.8215	.9117	.8698	.7895	.7208	.7897	.9078	.8171	.8122	.8510
Llama 8B-1B	SLM w/o	.6779	.6595	.6063	.6764	.7297	.7159	.6378	.7915	.5215	.5084	.5119	.5064	.6850	.6581	.6039	.6842
	SLM w/	.8070	.7915	.7161	.8083	.8604	.8529	.7805	.8866	.7115	.7291	.6023	.7263	.8134	.7897	.7159	.8049
	LLM w/o	.7227	.7110	.6895	.7169	.7476	.7355	.6689	.8021	.6044	.5909	.6574	.5967	.7405	.7137	.6903	.7306
	CoSteer	.8968	.8571	.7993	.8735	.9173	.8788	.8297	.9235	.8499	.8408	.6951	.8266	.9004	.8562	.8008	.8648
	LLM w/	.8951	.8375	.8121	.8603	.9030	.8556	.8108	.9004	.8634	.7974	.7410	.8101	.8860	.8285	.7997	.8496
Qwen 0.6B-8B	SLM w/o	.5844	.5477	.5984	.5945	.6498	.6291	.6573	.7194	.4014	.3648	.5039	.4144	.6485	.6018	.6327	.6463
	SLM w/	.6255	.6428	.6099	.6313	.7299	.7302	.6614	.7638	.4928	.5277	.4942	.4968	.7063	.6721	.6343	.6826
	LLM w/o	.7819	.7783	.7084	.7771	.8057	.8037	.6920	.8448	.6536	.6504	.6675	.6461	.7974	.7796	.7038	.7765
	CoSteer	.7849	.7806	.7046	.7757	.8058	.8117	.6901	.8477	.6582	.6549	.6647	.6506	.7986	.7819	.7058	.7853
	LLM w/	.8928	.8037	.7842	.8629	.9076	.8303	.8141	.9020	.8612	.7796	.6952	.8004	.8964	.7989	.7915	.8555
Qwen 0.6B-32B	SLM w/o	.5844	.5477	.5984	.5945	.6498	.6291	.6573	.7194	.4014	.3648	.5039	.4144	.6485	.6018	.6327	.6463
	SLM w/	.6255	.6428	.6099	.6313	.7299	.7302	.6614	.7638	.4928	.5277	.4942	.4968	.7063	.6721	.6343	.6826
	LLM w/o	.8103	.8084	.6991	.7922	.8179	.8237	.6948	.8443	.7084	.7130	.6863	.7010	.8077	.7948	.6935	.7758
	CoSteer	.8083	.8152	.6984	.7923	.8145	.8191	.6752	.8436	.7152	.7226	.6870	.7020	.8053	.7962	.6884	.7779
	LLM w/	.9121	.8147	.8294	.8722	.9094	.8352	.8298	.8936	.9097	.7886	.7364	.8327	.9069	.7996	.8204	.8629

Table 5: Results on all four preference alignment datasets with our proposed CoSteer and other settings. **Bold** values indicate that our method outperformed the three baseline methods we are comparing against. Note that results from the large model with context are marked in gray, as this scenario does not align with our task setting.

to time constraints, we randomly sampled 10 articles from each of the three LongLaMP datasets. We then invited five PhD candidates with strong NLP backgrounds to act as evaluators. Following the evaluation criteria from Cogensis, they rated the generated texts from all models on two aspects: **Overall Quality (OvI)** and **Personalization (Per)**, using a 1–5 scale. The average scores are presented in Table 7.

The results from our human evaluation corroborate the findings from our automated metrics. In all three tasks, **CoSteer** significantly improves personalization scores over the LLM w/o baseline and achieves an overall quality that is highly competitive with the full LLM w/ oracle. This demonstrates the effectiveness of our method in generating high-quality, personalized text that aligns with human judgment.

B.7 Prompts for LLM-as-a-Judge

We use exactly the same prompt as Cogensis (Zhang et al., 2024b) to have the gpt-4o evaluate the generated content. The specific prompts are shown in Figure 8 and Figure 9.

B.8 Baseline Implementation Details

We compare our proposed method against three representative baseline approaches:

CoS / LA Context Steering (CoS) (He et al., 2025) and Linear Alignment (LA) (Gao et al., 2024) are conceptually equivalent methods. They steer the generation of a base model by modifying its output log-probabilities. The final log-probability for a token x_i is calculated by adding a scaled difference term to the base model’s prediction, which is conditioned on no context ($\emptyset$). The formulation is as follows:

$$\log P'(x_i) = \log P(x_i | \emptyset, \mathcal{P}) + \lambda \cdot \left(\log P(x_i | C, \mathcal{P}) - \log P(x_i | \emptyset, \mathcal{P}) \right) \quad (17)$$

where $\mathcal{P}$ represents the base model parameters, C is the provided context, and λ is a scaling hyperparameter. Following the implementations in the original works, we searched for the optimal λ from the set $\{1.5, 2.0, 2.5\}$ and report the best-performing result for each task.

ModelsSetting	Cogenesis		Abstract Generation			Review Writing			Topic Writing			Pref Align				
	Ovl	Per	R-1	R-L	MET	R-1	R-L	MET	R-1	R-L	MET	creative	verbose	concise	uplifting	
Qwen 7B-1.5B	SLM w/o	6.63 $\pm$ 1.70	6.21 $\pm$ 1.88	36.48 $\pm$ 7.50	17.74 $\pm$ 3.85	27.57 $\pm$ 5.44	20.40 $\pm$ 8.88	10.39 $\pm$ 3.71	10.39 $\pm$ 6.07	25.21 $\pm$ 9.55	11.09 $\pm$ 3.24	17.51 $\pm$ 8.08	.5441 $\pm$.13	.5154 $\pm$.11	.6104 $\pm$.09	.5632 $\pm$.06
	SLM w/	7.81 $\pm$ 1.24	7.63 $\pm$ 1.43	39.75 $\pm$ 9.83	22.03 $\pm$ 9.83	27.76 $\pm$ 10.03	23.08 $\pm$ 9.95	12.40 $\pm$ 7.15	12.94 $\pm$ 8.90	22.89 $\pm$ 9.19	11.46 $\pm$ 6.30	17.12 $\pm$ 8.45	.6214 $\pm$.10	.6998 $\pm$.08	.7102 $\pm$.13	.6594 $\pm$.05
	LLM w/o	8.00 $\pm$ 1.18	7.63 $\pm$ 1.43	39.81 $\pm$ 6.86	20.53 $\pm$ 4.32	25.56 $\pm$ 7.11	30.15 $\pm$ 6.64	14.04 $\pm$ 2.86	17.71 $\pm$ 6.54	27.64 $\pm$ 8.70	11.93 $\pm$ 2.48	21.49 $\pm$ 5.51	.7058 $\pm$.10	.6931 $\pm$.08	.7039 $\pm$.13	.7183 $\pm$.07
	CoSteer	8.44* $\pm$ 1.31	8.50* $\pm$ 1.20	42.98* $\pm$ 8.94	23.61* $\pm$ 7.09	28.20* $\pm$ 8.88	32.72* $\pm$ 9.67	15.92* $\pm$ 9.02	20.36* $\pm$ 10.09	25.93 $\pm$ 10.13	12.38* $\pm$ 9.78	22.84* $\pm$ 10.41	.7915* $\pm$.06	.7608* $\pm$.07	.7404* $\pm$.08	.7885* $\pm$.14
	LLM w/	8.62 $\pm$ 0.63	8.61 $\pm$ 0.58	44.50 $\pm$ 9.77	24.63 $\pm$ 9.14	31.15 $\pm$ 9.34	33.83 $\pm$ 5.44	15.55 $\pm$ 2.41	22.42 $\pm$ 4.76	28.82 $\pm$ 10.76	13.77 $\pm$ 9.37	23.44 $\pm$ 8.29	.8884 $\pm$.07	.8076 $\pm$.05	.7693 $\pm$.07	.8414 $\pm$.16
Qwen 32B-7B	SLM w/o	8.00 $\pm$ 1.18	7.63 $\pm$ 1.43	39.81 $\pm$ 6.86	20.53 $\pm$ 4.32	25.56 $\pm$ 7.11	30.15 $\pm$ 6.64	14.04 $\pm$ 2.86	17.71 $\pm$ 6.54	27.64 $\pm$ 8.70	11.93 $\pm$ 2.48	21.49 $\pm$ 5.51	.7058 $\pm$.09	.6931 $\pm$.09	.7039 $\pm$.08	.7183 $\pm$.15
	SLM w/	8.62 $\pm$ 0.63	8.61 $\pm$ 0.58	44.50 $\pm$ 9.77	24.63 $\pm$ 9.14	31.15 $\pm$ 9.34	33.83 $\pm$ 5.44	15.55 $\pm$ 2.41	22.42 $\pm$ 4.76	28.82 $\pm$ 11.76	13.77 $\pm$ 9.37	23.44 $\pm$ 8.29	.8884 $\pm$.05	.8076 $\pm$.06	.7693 $\pm$.12	.8414 $\pm$.05
	LLM w/o	8.12 $\pm$ 1.01	7.87 $\pm$ 1.16	40.66 $\pm$ 6.79	21.02 $\pm$ 4.26	26.61 $\pm$ 6.83	32.21 $\pm$ 6.32	14.44 $\pm$ 2.51	19.61 $\pm$ 6.44	28.82 $\pm$ 8.53	12.20 $\pm$ 2.68	21.16 $\pm$ 6.11	.7020 $\pm$.14	.6873 $\pm$.06	.7225 $\pm$.10	.7146 $\pm$.13
	CoSteer	8.78* $\pm$ 0.81	8.64* $\pm$ 0.79	45.41* $\pm$ 9.79	26.04* $\pm$ 9.99	33.52* $\pm$ 9.64	34.88* $\pm$ 8.80	15.89* $\pm$ 4.72	26.51* $\pm$ 7.23	30.10* $\pm$ 12.06	14.52* $\pm$ 10.76	24.20* $\pm$ 11.26	.8589 $\pm$.06	.8532* $\pm$.08	.7274 $\pm$.06	.8579* $\pm$.04
	LLM w/	8.83 $\pm$ 0.48	8.76 $\pm$ 0.51	43.33 $\pm$ 9.16	23.47 $\pm$ 6.40	30.10 $\pm$ 7.36	34.65 $\pm$ 5.63	15.74 $\pm$ 3.83	22.77 $\pm$ 5.48	30.73 $\pm$ 10.83	14.20 $\pm$ 8.45	24.25 $\pm$ 8.67	.9017 $\pm$.03	.8193 $\pm$.07	.7912 $\pm$.12	.8538 $\pm$.06
Llama 8B-1B	SLM w/o	7.04 $\pm$ 1.65	6.55 $\pm$ 1.81	33.20 $\pm$ 8.60	18.20 $\pm$ 3.78	28.55 $\pm$ 4.84	31.75 $\pm$ 5.95	14.92 $\pm$ 2.37	19.78 $\pm$ 5.53	20.81 $\pm$ 10.10	10.21 $\pm$ 3.22	17.30 $\pm$ 8.29	.6535 $\pm$.12	.6355 $\pm$.11	.5900 $\pm$.10	.6646 $\pm$.12
	SLM w/	7.69 $\pm$ 1.40	7.52 $\pm$ 1.39	39.81 $\pm$ 9.63	21.53 $\pm$ 4.95	30.11 $\pm$ 6.45	32.36 $\pm$ 6.59	15.02 $\pm$ 2.70	22.06 $\pm$ 4.80	20.17 $\pm$ 10.43	10.58 $\pm$ 5.99	18.64 $\pm$ 8.61	.7981 $\pm$.11	.7908 $\pm$.08	.7037 $\pm$.13	.8065 $\pm$.10
	LLM w/o	7.69 $\pm$ 1.38	7.13 $\pm$ 1.62	39.33 $\pm$ 8.11	20.69 $\pm$ 3.90	29.41 $\pm$ 5.32	34.58 $\pm$ 5.19	15.32 $\pm$ 2.43	22.10 $\pm$ 5.54	26.93 $\pm$ 10.11	12.35 $\pm$ 3.40	21.82 $\pm$ 6.47	.7038 $\pm$.09	.6878 $\pm$.06	.6765 $\pm$.09	.7116 $\pm$.09
	CoSteer	7.29 $\pm$ 1.42	7.73* $\pm$ 1.03	41.28* $\pm$ 8.36	24.97* $\pm$ 8.10	31.19* $\pm$ 6.42	31.68 $\pm$ 6.33	13.68 $\pm$ 4.63	24.57* $\pm$ 6.93	26.11 $\pm$ 11.18	12.00 $\pm$ 6.74	23.69* $\pm$ 8.13	.8911* $\pm$.05	.8582* $\pm$.08	.7812* $\pm$.12	.8721* $\pm$.10
	LLM w/	8.61 $\pm$ 0.66	8.44 $\pm$ 0.80	43.91 $\pm$ 9.69	23.93 $\pm$ 6.44	32.01 $\pm$ 7.97	36.39 $\pm$ 5.19	15.95 $\pm$ 2.50	23.56 $\pm$ 4.80	30.54 $\pm$ 10.43	14.02 $\pm$ 6.15	23.81 $\pm$ 7.39	.8869 $\pm$.06	.8298 $\pm$.05	.7909 $\pm$.07	.8551 $\pm$.06
Qwen 8B-0.6B	SLM w/o	6.84 $\pm$ 1.70	6.64 $\pm$ 1.80	37.13 $\pm$ 7.17	20.41 $\pm$ 4.90	21.00 $\pm$ 6.91	24.41 $\pm$ 7.88	12.83 $\pm$ 3.69	12.39 $\pm$ 5.68	24.19 $\pm$ 9.46	11.88 $\pm$ 3.88	15.74 $\pm$ 6.38	.5710 $\pm$.13	.5359 $\pm$.04	.5981 $\pm$.10	.5937 $\pm$.11
	SLM w/	7.85 $\pm$ 1.49	7.79 $\pm$ 1.42	42.48 $\pm$ 8.39	23.25 $\pm$ 6.21	26.58 $\pm$ 8.29	25.04 $\pm$ 10.20	13.36 $\pm$ 7.38	13.02 $\pm$ 8.71	26.17 $\pm$ 11.92	13.46 $\pm$ 9.01	18.40 $\pm$ 10.13	.6386 $\pm$.12	.6432 $\pm$.14	.6000 $\pm$.14	.6436 $\pm$.12
	LLM w/o	8.27 $\pm$ 1.05	7.99 $\pm$ 1.25	40.76 $\pm$ 7.06	21.23 $\pm$ 4.37	25.77 $\pm$ 6.75	30.89 $\pm$ 6.69	14.47 $\pm$ 2.85	16.98 $\pm$ 5.79	28.47 $\pm$ 9.30	12.85 $\pm$ 2.94	21.49 $\pm$ 6.65	.7597 $\pm$.15	.7530 $\pm$.14	.6929 $\pm$.15	.7611 $\pm$.07
	CoSteer	8.62* $\pm$ 0.90	8.65* $\pm$ 1.12	41.11 $\pm$ 8.51	22.36 $\pm$ 6.21	25.86 $\pm$ 7.98	32.24* $\pm$ 7.22	14.84* $\pm$ 3.02	18.47* $\pm$ 6.15	29.03* $\pm$ 11.07	13.61* $\pm$ 7.06	22.98* $\pm$ 8.47	.7619* $\pm$.06	.7573* $\pm$.08	.6913 $\pm$.05	.7648* $\pm$.11
	LLM w/	8.96 $\pm$ 0.46	8.96 $\pm$ 0.36	43.99 $\pm$ 8.46	23.69 $\pm$ 6.42	29.48 $\pm$ 7.76	35.42 $\pm$ 5.45	15.71 $\pm$ 2.55	21.59 $\pm$ 4.96	31.48 $\pm$ 13.00	15.34 $\pm$ 11.50	24.78 $\pm$ 11.46	.8895 $\pm$.06	.8031 $\pm$.12	.7713 $\pm$.09	.8552 $\pm$.05
Qwen 32B-0.6B	SLM w/o	6.84 $\pm$ 1.70	6.64 $\pm$ 1.80	37.13 $\pm$ 7.17	20.41 $\pm$ 4.90	21.00 $\pm$ 6.91	24.41 $\pm$ 7.88	12.83 $\pm$ 3.69	12.39 $\pm$ 5.68	24.19 $\pm$ 9.46	11.88 $\pm$ 3.88	15.74 $\pm$ 6.38	.5710 $\pm$.06	.5359 $\pm$.10	.5981 $\pm$.12	.5937 $\pm$.06
	SLM w/	7.85 $\pm$ 1.49	7.79 $\pm$ 1.42	42.48 $\pm$ 8.39	23.25 $\pm$ 6.21	26.58 $\pm$ 8.29	25.04 $\pm$ 10.20	13.36 $\pm$ 7.38	13.02 $\pm$ 8.71	26.17 $\pm$ 11.92	13.46 $\pm$ 9.01	18.40 $\pm$ 10.13	.6386 $\pm$.08	.6432 $\pm$.11	.6000 $\pm$.10	.6436 $\pm$.12
	LLM w/o	8.31 $\pm$ 0.92	8.22 $\pm$ 1.12	41.05 $\pm$ 6.95	21.88 $\pm$ 4.54	26.23 $\pm$ 7.01	30.23 $\pm$ 6.94	14.08 $\pm$ 3.03	16.41 $\pm$ 5.75	29.01 $\pm$ 10.00	12.51 $\pm$ 3.06	21.44 $\pm$ 6.23	.7861 $\pm$.10	.7850 $\pm$.08	.6934 $\pm$.10	.7783 $\pm$.06
	CoSteer	8.52* $\pm$ 0.80	8.56* $\pm$ 0.87	43.74* $\pm$ 9.51	23.69* $\pm$ 8.47	30.97* $\pm$ 9.74	30.89* $\pm$ 9.37	14.14* $\pm$ 4.89	18.87* $\pm$ 7.62	26.75 $\pm$ 12.75	13.17 $\pm$ 10.14	21.32 $\pm$ 8.47	.7858 $\pm$.11	.7883* $\pm$.09	.6873 $\pm$.14	.7790* $\pm$.07
	LLM w/	9.09 $\pm$ 0.48	9.03 $\pm$ 0.29	44.46 $\pm$ 8.08	23.95 $\pm$ 5.47	30.15 $\pm$ 6.98	35.12 $\pm$ 6.20	15.60 $\pm$ 2.70	20.65 $\pm$ 5.45	32.63 $\pm$ 12.44	15.37 $\pm$ 10.48	25.07 $\pm$ 10.63	.9095 $\pm$.05	.8095 $\pm$.08	.8040 $\pm$.13	.8654 $\pm$.06

Table 6: Main Results with Mean and Standard Deviation. **Bold** entries indicate that CoSteer outperforms the three baselines (SLM w/o, SLM w/, LLM w/o). An asterisk (*) denotes that this improvement is statistically significant (paired t-test, $p < 0.05$). Gray values represent the privacy-violating near-upper-bound performance, yet underlined CoSteer values surpass these incompatible references.

Table 7: Human evaluation results on a 1-5 scale (higher is better). Scores are reported as **Overall Quality (Ovl) / Personalization (Per)**. Our method, CoSteer, achieves personalization scores comparable to the LLM w/ oracle while maintaining high overall quality.

Method	Abstract (Ovl / Per)	Review (Ovl / Per)	Writing (Ovl / Per)
SLM w/o	3.85 / 3.70	3.45 / 3.25	3.20 / 3.05
SLM w/	3.95 / 3.85	3.70 / 3.75	3.35 / 3.50
LLM w/o	4.30 / 3.80	4.15 / 3.40	3.95 / 3.55
CoSteer	4.30 / 3.90	3.90 / 3.80	4.10 / 4.15
LLM w/	4.45 / 4.00	4.30 / 4.05	4.15 / 4.25

Supervised Fine-Tuning (SFT) To create a strong task-specific baseline, we fine-tuned the small language model using Low-Rank Adaptation (LoRA). For each of our three datasets, we randomly sampled 1,000 data points for training. The model was then trained for three epochs. Key training parameters included a learning rate of 1×10^{-4} with a cosine scheduler, a maximum sequence length of 1024, a weight decay of 0.05, and bf16 precision for efficiency. We utilized packing

to handle variable-length sequences effectively.

Proxy-Tuning This method (Liu et al., 2024) enhances a large base model by incorporating signals from a smaller, specialized proxy model. The final probability distribution is calculated by adjusting the logits of the large model. We adapt the original formula to better reflect the roles of the different models in our setup:

$$p(x_t | x_{<t}) = \text{softmax} \left[s_{\text{LLM}}(x_t | x_{<t}) + s_{\text{SLM-SFT}}(x_t | x_{<t}) - s_{\text{SLM-base}}(x_t | x_{<t}) \right]. \quad (18)$$

Here, s_{LLM} represents the logits from the large, pre-trained language model. The term $s_{\text{SLM-SFT}}$ refers to the logits from the small language model that has been fine-tuned (the SFT baseline), and $s_{\text{SLM-base}}$ refers to the logits from the original, pre-trained small language model.

The comparison results are shown in Table 8.

Models	Method	Abstract			Review			Writing		
		R-1	R-L	MET	R-1	R-L	MET	R-1	R-L	MET
Qwen 7B-1.5B	LA/CS	<u>41.49</u>	25.57	29.96	26.60	13.68	17.59	24.09	<u>12.32</u>	19.94
	SFT	40.71	21.85	27.00	<u>31.55</u>	12.91	<u>20.23</u>	25.53	10.80	19.35
	Proxy tuning	40.76	21.37	25.95	29.54	<u>14.16</u>	16.79	27.13	12.11	<u>21.38</u>
	CoSteer	42.98	<u>23.61</u>	<u>28.20</u>	32.72	15.92	20.36	<u>25.93</u>	12.38	22.84
Qwen 32B-7B	LA/CS	44.28	<u>25.85</u>	<u>32.98</u>	<u>34.14</u>	<u>15.46</u>	<u>26.38</u>	<u>29.22</u>	<u>14.03</u>	<u>23.66</u>
	SFT	43.59	25.48	32.43	31.94	11.84	21.00	28.83	10.63	21.73
	Proxy tuning	<u>44.61</u>	25.12	31.97	32.03	14.80	19.01	28.75	12.70	21.49
	CoSteer	45.41	26.04	33.52	34.88	15.89	26.51	30.10	14.52	24.20
Llama 8B-1B	LA/CS	37.24	20.70	<u>30.78</u>	27.35	11.93	23.54	19.28	9.30	18.26
	SFT	<u>40.62</u>	26.67	25.64	29.93	10.17	21.51	26.11	8.99	<u>22.56</u>
	Proxy tuning	39.55	20.78	29.60	<u>30.50</u>	<u>12.29</u>	<u>21.81</u>	26.00	12.29	21.65
	CoSteer	41.28	<u>24.97</u>	31.19	31.68	13.68	24.57	26.11	<u>12.00</u>	23.69
Qwen 8B-0.6B	LA/CS	31.42	16.78	22.10	28.37	12.92	15.94	25.31	11.92	18.96
	SFT	38.38	<u>21.88</u>	26.03	30.53	<u>14.75</u>	19.91	25.20	12.82	18.83
	Proxy tuning	<u>40.60</u>	21.14	25.68	<u>30.97</u>	14.48	16.94	<u>28.84</u>	<u>12.89</u>	<u>21.64</u>
	CoSteer	41.11	22.36	<u>25.86</u>	32.24	14.84	<u>18.47</u>	29.03	13.61	22.98
Qwen 32B-0.6B	LA/CS	31.42	16.78	22.10	28.37	12.92	15.94	25.31	11.92	18.96
	SFT	38.38	<u>21.88</u>	26.03	<u>30.53</u>	14.75	<u>18.91</u>	25.20	<u>12.82</u>	18.83
	Proxy tuning	<u>41.17</u>	<u>21.88</u>	<u>26.40</u>	30.20	<u>14.14</u>	16.27	<u>26.08</u>	12.61	21.67
	CoSteer	43.74	23.69	30.97	30.89	<u>14.14</u>	18.87	26.75	13.17	<u>21.32</u>

Table 8: Performance on LongLamp benchmark across 5 model pairs. Best results are marked in **bold** and second-best are underlined.

B.9 Parameter Sensitivity Analysis

Results are shown in Figure 3. Below, we analyze each parameter.

- **Parameter α and β :** As defined in Equation 4, α and β adjust the influence of each optimization component. α controls the impact of the difference between the current policy and the initial policy, while β scales the delta signals. We conducted experiments with values ranging from 0.1 to 5.0. As shown in the first two tables of the figure, small α and β coefficients below 1.0 attenuate personalization signals, whereas values exceeding 3.0 compromise the model’s fundamental capabilities through over-alignment, leading to a noticeable decline in results. Therefore we adopt $\alpha = 2$ and $\beta = 1$ for performance.
- **Parameter λ and η :** These two learning dynamics control the learning rate and the degree of deviation from the initial policy. For the learning rate η , we conducted experiments with values ranging from 2 to 20. For η , we observe stable convergence from 2 to 10, followed by oscillatory behavior beyond 10. Parameter λ rapid performance gains from 0 to 2, and then transitions to a plateau phase be-

tween 2 and 5. Therefore, we set η to 10 and λ to 2.

- **Iteration step T :** We studied the impact of T within the range of 1 to 100. As shown in the results of the last table, there is a clear upward trend in performance from 1 to 20 iterations, after which the performance stabilizes. We thus establish $T = 20$ as the Pareto-optimal configuration balancing quality and latency.

B.10 Robustness to Noise

As a supplement to Section 5.1, we detail the experimental setup and results for our robustness tests.

B.10.1 Robustness to Realistic Noise (Weaker Retriever)

Experimental Setup In many real-world systems, the retrieval component may not be state-of-the-art. To simulate this “realistic noise”, where retrieved context is topically relevant but potentially less precise, we replaced the powerful bge-m3 retriever with the classic, non-neural BM25 retriever. This setup tests whether **CoSteer** is overly dependent on a high-quality retriever or if it can adapt to a more common, weaker signal. We use Qwen7B-1.5B pair to conduct the experiment.

Results and Analysis The results are presented in Table 9. Even with the noisier context pro-

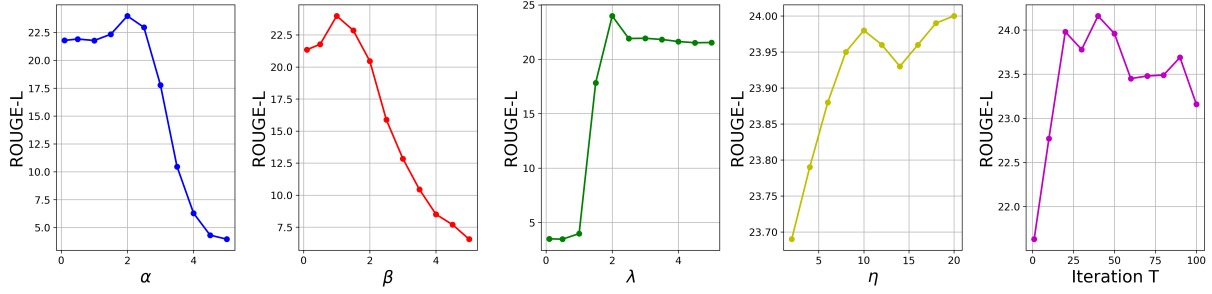


Figure 3: Effect of α , β , η , λ and iteration step T on the Abstract Generation dataset using Qwen 7B-1.5B. Metric: ROUGE-L.

vided by BM25, **CoSteer** demonstrates strong performance, achieving the second-best results across nearly all metrics, only behind the full LLM w/ oracle. For instance, in the Abstract generation task, **CoSteer** (43.08 R-1) significantly outperforms both the SLM w/ (39.51 R-1) and the strong LLM w/o (39.81 R-1) baselines. This confirms that our framework is not brittle and can effectively leverage imperfect but relevant local information, a critical capability for practical deployment.

B.10.2 Robustness to Purely Irrelevant (Adversarial) Noise

Experimental Setup To push the limits of our framework, we designed a more adversarial scenario to test a potential failure mode: providing the model with context that is completely irrelevant to the task. For example, when generating an Abstract, we supplied context examples drawn from the Review task. This setup evaluates whether the model can ignore distracting information or if it gets catastrophically misled. We use Qwen7B-1.5B pair to conduct the experiment.

Results and Analysis Table 10 shows the outcomes of this stress test. As expected, the performance of all models degrades compared to using relevant context. However, the key observation is in the *manner* of degradation. **CoSteer** (e.g., 39.59 R-1 on Abstract) experiences a controlled performance drop but still comfortably outperforms the specialized SLM w/ (34.91 R-1). It does not collapse. This graceful degradation suggests that the strong inductive bias from the frozen base LLM acts as a safeguard, preventing the steering mechanism from being entirely derailed by nonsensical context. The model learns to rely more on its pre-trained knowledge when context is useless, showcasing a desirable level of robustness.

B.11 Results and Implementation of Cross-Architecture Collaboration

The experiments were conducted using Llama-3.1-8B as the LLM and Qwen2.5-1.5B as the SLM. Results are shown in Table 11. Below we detail the two strategies we implemented to facilitate collaboration between models with different architectures and tokenizers, as discussed in Section 5.2.

B.11.1 Vocabulary Mapping (CoSteer_map)

Principle The vocabulary mapping approach establishes a shared communication channel by restricting fusion to the intersection of the two models’ vocabularies, allowing agreement on a common set of semantic units.

Implementation Our implementation follows a three-step process at each generation step:

1. **Vocabulary Intersection:** Before generation begins, we create a mapping between the two models. We extract the token strings from both tokenizers and find their intersection. Two tensors, `llm_intersect_ids` and `slm_intersect_ids`, are then created to store the corresponding token IDs for this shared vocabulary, ensuring a deterministic alignment.
2. **Logit Projection:** At each decoding step, we obtain the native logit outputs from both the LLM and SLM. Using `torch.index_select`, we project these full-vocabulary logits onto the smaller, shared vocabulary space defined by the intersection.
3. **Optimization and Remapping:** The CoSteer optimization is applied in this shared space. The resulting token is mapped back to each model’s native ID space via the pre-computed tensors and appended to their respective input sequences.

	Abstract			Review			Writing		
	R-1	R-L	MET	R-1	R-L	MET	R-1	R-L	MET
SLM w/o	36.48	17.74	27.57	20.40	10.39	10.39	25.11	11.09	17.51
SLM w/	39.51	21.92	27.74	23.42	12.00	13.14	24.15	12.32	18.22
LLM w/o	39.81	20.53	25.56	30.15	14.04	17.71	27.64	11.93	21.49
CoSteer	43.08	23.49	28.08	31.95	15.36	19.45	25.47	12.46	22.83
LLM w/	44.17	24.55	30.83	33.70	15.42	22.47	28.90	13.89	23.84

Table 9: Robustness to a weaker retriever (BM25). Even with this noisier context, CoSteer consistently outperforms both the SLM w/ and the LLM w/o baselines, demonstrating its robustness in practical scenarios.

	Abstract			Review			Writing		
	R-1	R-L	MET	R-1	R-L	MET	R-1	R-L	MET
SLM w/o	36.48	17.74	27.57	20.40	10.39	10.39	25.11	11.09	17.51
SLM w/	34.91	18.31	24.65	22.15	11.56	12.25	22.68	9.71	18.03
LLM w/o	39.81	20.53	25.56	30.15	14.04	17.71	27.64	11.93	21.49
CoSteer	39.59	21.17	24.16	30.45	14.11	17.76	24.71	10.80	20.19
LLM w/	38.43	20.72	23.65	31.69	14.49	21.01	25.48	11.22	20.65

Table 10: Robustness to purely irrelevant (adversarial) noise. We tested a failure mode by providing irrelevant context (e.g., Review examples for the Abstract task). The framework degrades gracefully rather than failing catastrophically, still outperforming the local SLM.

While conceptually simple and effective, a key limitation of this approach is that the intersection can be small for models with dissimilar tokenizers, potentially limiting the expressivity of the generation.

B.11.2 Byte-level Fusion (CoSteer_byte)

Principle To overcome the limitations of vocabulary mapping, CoSteer_byte operates entirely in a shared byte space. Instead of aligning token IDs, it projects each model’s token-level logit distribution into a common 257-dimensional space (Hayase et al., 2025): 256 dimensions for possible next bytes (0–255) and one special “commit” dimension (byte 256) indicating that the current token is complete. This enables direct byte-level probabilistic fusion of LLM and SLM outputs.

Implementation The byte-level strategy integrates with our optimization as follows:

1. **Byte Projection:** For each model, we map its token-level logits to a byte-level log-probability distribution over the 257-dimensional space. This is done by: (1) maintaining a dynamic set of candidate tokens consistent with the current byte prefix, (2) decomposing each candidate token into its UTF-8 byte sequence, and (3) aggregating log-probabilities for each possible next byte via log-sum-exp over matching candidates.
2. **Optimization and Byte Sampling:** CoSteer fuses the three byte-level distributions into a

single policy, from which the next byte is sampled. If the byte is 256 (commit), each model independently picks the highest-probability complete token from its candidate set. Despite differing token IDs due to tokenizer mismatch, these tokens decode to the same UTF-8 string, ensuring semantic alignment. Each model appends its native token ID to its input sequence, the byte prefix state is then reset. Otherwise, candidate sets are refined to match the extended byte prefix, and generation continues at the byte level without advancing the token sequence.

This approach requires no vocabulary overlap and supports arbitrary model pairs. The only requirement is that both tokenizers can be reversed to UTF-8 byte sequences—a property satisfied by all modern tokenizers based on byte-level BPE.

B.12 Implementation and Analysis of CoSteer Variants

This section provides a detailed description of the **LightCoSteer** and **AdaCoSteer** variants, followed by a comprehensive analysis of their performance and efficiency trade-offs.

B.12.1 LightCoSteer: Single-Step Steering

Motivation and Implementation The primary source of computational overhead in the full **CoSteer** framework is the iterative optimization process within the FTRL algorithm. **LightCoSteer** is

	Abstract			Review			Writing		
	R-1	R-L	MET	R-1	R-L	MET	R-1	R-L	MET
SLM w/o	36.48	17.64	27.57	20.40	10.39	10.39	25.21	11.09	17.51
SLM w/	39.75	22.03	27.76	23.08	12.40	12.94	22.89	11.46	17.12
LLM w/o	39.33	20.69	29.41	34.58	15.32	22.10	26.93	12.35	21.82
CoSteer_map	41.82	23.40	31.79	33.72	15.63	23.22	25.01	11.78	21.10
CoSteer_byte	42.84	22.69	32.03	33.58	15.24	23.50	23.29	10.94	20.14
LLM w/	43.91	23.93	32.01	36.39	15.95	23.56	30.54	14.02	23.81

Table 11: Performance of **CoSteer** in a cross-architecture setting, using Llama-3.1-8B as the LLM and Qwen2.5-1.5B as the SLM. We compare two vocabulary-agnostic strategies: vocabulary mapping (CoSteer_map) and a more universal byte-level fusion (CoSteer_byte). Both methods enable effective collaboration, validating our framework’s generalization capability across model families.

designed to eliminate this overhead entirely. We achieve this by setting the number of iterations to one ($T = 1$). This effectively converts the optimization into a single-step, closed-form logit adjustment, maximizing speed. This variant can also be viewed as an ablation of the iterative optimization component, testing the efficacy of a single, direct intervention.

Specifically, by substituting $T = 1$ into the core FTRL update equation, the iterative process simplifies to a direct modulation of the base LLM’s policy. The resulting policy for selecting an action (token) a is equivalent to:

$$\pi(a) \propto \exp \left(\log \pi_0(a) + \beta^* \left(\log \pi_{\text{pers}}^*(a) - \log \pi_{\text{base}}^*(a) \right) \right). \quad (19)$$

where π_0 is the base LLM policy, π_{pers}^* and π_{base}^* are the SLM policies with and without context respectively, and β^* is a weighting hyperparameter.

B.12.2 AdaCoSteer: Adaptive Steering

Motivation and Implementation While **LightCoSteer** addresses computational load, it does not reduce the number of communication rounds between the local SLM and the remote LLM, as every token still requires steering. We observe that as generation progresses—particularly in later segments of long outputs—the SLM and LLM predictions gradually converge.

Motivated by this, **AdaCoSteer** implements an adaptive strategy to reduce unnecessary communication. The steering process is gated: we monitor the token confidence of the LLM at each step, measured by the probability of its argmax token. If this confidence score exceeds a predefined threshold τ for k consecutive steps, we deactivate **CoSteer**. The framework then switches to a vanilla generation mode, allowing the LLM to autonomously

complete the sequence. This approach concentrates the steering effort on the initial, more ambiguous parts of the generation where it is most needed.

B.12.3 Performance

We present the performance and efficiency results of the two variants compared to the full **CoSteer** framework in Table 12, Table 13, and Table 14.

Effectiveness Analysis Table 12 shows the performance trade-offs. The full **CoSteer** framework consistently achieves the highest scores across all tasks. **LightCoSteer** follows closely, demonstrating that a single-step adjustment retains a significant portion of the framework’s benefits. **AdaCoSteer** experiences a slightly larger performance drop, which is expected as it deliberately stops steering in later, potentially less critical, generation stages.

B.12.4 Efficiency and Overhead Analysis

Empirical Results Table 13 quantifies the throughput (tokens/second) of our proposed variants, while Table 14 compares their computational cost (FLOPS) against baselines. The empirical results reveal several key findings.

First, as shown in Table 13, **LightCoSteer** (13.73 tok/s), which performs a single-step adjustment, is markedly faster than the fully iterative **CoSteer** (9.44 tok/s). More strikingly, **AdaCoSteer** (20.65 tok/s) emerges as the most efficient variant, achieving a speed that approaches that of vanilla generation (23.88 tok/s). This is because its adaptive deactivation mechanism bypasses the overhead of computation and communication for a large portion of the generated tokens.

Second, the FLOPS analysis in Table 14 highlights the core architectural advantage of our approach. Even the most intensive variant, **CoSteer**, remains far more computationally efficient than the naive baseline of sending the full context to the

	Abstract			Review			Writing		
	R-1	R-L	MET	R-1	R-L	MET	R-1	R-L	MET
AdaCoSteer	40.97	22.04	25.96	29.94	14.22	17.15	26.48	12.12	22.24
LightCoSteer	41.79	21.86	26.44	31.46	14.80	18.31	27.33	12.19	21.89
CoSteer	42.98	23.61	28.20	32.72	15.92	20.36	25.93	12.38	22.84

Table 12: Performance comparison of different CoSteer framework variants.

Method	Time Complexity	Speed (tok/s)
Vanilla Gen.	$L(n)$	23.88
AdaCoSteer	$L(n) + C(t_c) + T(c)$	20.65
LightCoSteer	$L(n) + C(n) + T(n)$	13.73
CoSteer	$L(n) + C(t_n) + T(n)$	9.44

Table 13: Inference speed and time complexity analysis for CoSteer and its variants.

Method	C=10	C=100	C=1000
LLM w/o	0.87	0.87	0.87
LLM w/	0.96	1.74	9.98
CoSteer	1.21	1.35	2.98

Table 14: Computational cost (TFLOPs) as a function of context length (C).

LLM (‘LLM w/’). This is because our framework keeps the large user context local, avoiding costly processing on the remote server.

Analysis of Latency Components To clearly identify the sources of overhead and justify our design choices, we provide a detailed breakdown of the wall-clock time per token in Figure 4. This visualization highlights the **asynchronous pipelining** workflow inherent to CoSteer.

As illustrated in Figure 4, the generation process is split into two parallel streams once a token is sampled:

- **Stream 1 (Critical Path):** Involves uploading the token, Cloud LLM inference, and downloading logits. This path is dominated by network transmission ($\mathcal{T}(n) \approx 40\text{ms}$) and cloud inference ($\mathcal{L}(n) \approx 40\text{ms}$).
- **Stream 2 (Masked Path):** Simultaneously, the local SLM performs batched inference. Crucially, because the local NPU inference time ($\approx 30\text{ms}$) is masked by the longer duration of Stream 1, it does not impose a penalty on the total latency.

Based on this breakdown, we proposed **LightCoSteer** and **AdaCoSteer** to systematically address the actual unmasked bottlenecks on the critical path:

- **LightCoSteer** targets the *Optimization Bottleneck* (d). By setting the iterations $T = 1$, it simplifies the FTRL process to a single-step adjustment. This variant confirms that while the iterative optimization ($\approx 25\text{ms}$) adds overhead, it is manageable.

- **AdaCoSteer** targets the *Transmission Bottleneck* (b & e and together with d since no additional fusion are needed). Motivated by the observation that LLM and SLM predictions converge over time, it employs an adaptive termination strategy (Song et al., 2025). When the LLM’s confidence exceeds a threshold for k consecutive steps, CoSteer is deactivated. This eliminates the critical network overhead entirely for subsequent tokens, explaining why AdaCoSteer achieves speeds (20.65 tok/s) comparable to vanilla generation.

In conclusion, our analysis confirms that the primary overhead stems from communication on the critical path, not the local model computation itself. Adaptive strategies like **AdaCoSteer** effectively mitigate this bottleneck, offering a flexible balance between personalized generation and operational efficiency.

In summary, these variants form a clear and practical spectrum: **CoSteer** for maximum quality, **LightCoSteer** for a high-speed computational alternative, and **AdaCoSteer** for minimizing communication rounds and achieving the lowest latency.

C Broader Impact and Future Work

Broader Impact The primary contribution of our work lies in addressing the critical tension between the advancement of large-scale AI and the fundamental right to personal privacy. By enabling powerful, cloud-based LLMs to be personalized using sensitive data that never leaves the user’s device, the CoSteer framework offers a practical and scalable solution for **privacy-preserving AI**.

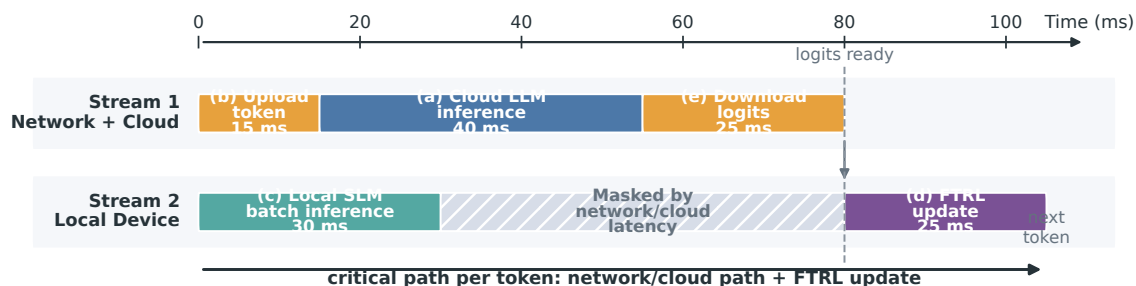


Figure 4: **Detailed breakdown of wall-clock time per token.** The system employs an **asynchronous pipelining** strategy: The Local SLM inference (Stream 2) is executed simultaneously with the Network/Cloud path (Stream 1). Since the local inference time ($\approx 30\text{ms}$) is typically shorter than the network round-trip ($\approx 40\text{ms}$) plus cloud inference ($\approx 40\text{ms}$), the local computational burden is effectively **masked** and does not affect the critical path latency. The primary bottlenecks are thus Network Transmission (b & e) and the local FTRL Optimization (d).

This aligns directly with societal demands for data sovereignty and user-centric control.

Furthermore, our approach represents a significant step forward in the field of **AI alignment**. Instead of relying on a single, universal alignment target, CoSteer facilitates a more dynamic and democratic form of alignment where model behavior is adapted in real-time to the unique context, preferences, and style of the individual. This fosters a safer and more beneficial human-AI interaction paradigm, reducing the risk of generating generic or contextually inappropriate content. By decentralizing the personalization process, our work contributes to a more equitable technological ecosystem where users can benefit from state-of-the-art AI without compromising their data.

Future Work The CoSteer framework establishes a versatile foundation for a new class of collaborative, privacy-aware generative models. Looking ahead, we plan to explore the broader potential of this paradigm.

D LLM Usage

The authors acknowledge the use of large language models (LLMs) during the composition of this paper. The models’ role was confined to that of an assistive tool for language refinement, including grammar correction, stylistic improvements, and punctuation adjustments. It is important to state that the LLMs did not contribute to the generation of any original scientific concepts, experimental methodologies, or the analysis and interpretation of results. The intellectual contributions are entirely those of the human authors.

Cogenesis

[Task]

Design an invitation for 'Homes for Humanity' charity event: Utilize your bi-monthly volunteering work and personal charm to craft an invitation.

[User Profile]

Age: 57 years

Name: Martin Reynolds

Occupation: Senior Real Estate Agent

Location: Charlotte, North Carolina

Personal traits: Detail-oriented; Personable; Strategic thinker; Avid golfer; Enjoys weekend DIY projects

Writing style: Clear and concise; Persuasive and sales-driven; Friendly, often includes anecdotes related to golf or DIY projects; Professional, with a touch of personal charm

Privacy info: Successfully renegotiated the lease terms for the company office last month; Celebrated 30th wedding anniversary with a surprise garden party; Volunteers bi-monthly at the local 'Homes for Humanity' charity; Recently started attending a beginner's pottery class on Sundays; Has a standing golf game every Wednesday afternoon with industry peers; Is secretly learning Spanish online to communicate with more clients; Adopts a new 2018 Real Estate Investment 2019 tip of the month for social media

Smart device usage: Email: Sent a client proposal at 8:45 AM - '184 Maple Drive Listing Proposal'; Calendar: Golf game scheduled for May 5, 1:30 PM with note - 'Meet Greg at Fairways Club'; Photos: Pictures of a kitchen renovation 'Before and After' at 3:52 PM; App: Logged into 'HouseFlipper' real estate investment app at 9:17 PM; Text Message: Received at 2:15 PM - 'Happy Anniversary! Table booked at La Trattoria.'; Voice Assistant: Asked for 'directions to the nearest hardware store' at 10:10 AM; Purchase: Ordered a 'Stainless Steel Faucet' from 'DIYHome' app at 7:45 PM; Notification: 'Your Spanish lesson starts in 5 minutes' at 7:55 AM; Search: Googled 'Energy-efficient home improvements' at 10:25 AM

Ai assistant usage: Composing a visually rich monthly newsletter for property investors; Creating client-specific follow-up 200bemails capturing unique details of each viewing; Generating personalized invitations for exclusive open houses and real estate events"

[writing history]

[query]:Draft a proposal for '184 Maple Drive' listing: Using the information from your morning email, flesh out a comprehensive proposal for the '184 Maple Drive' property.

[response]:"Dear Potential Investor, In the realm of real estate, opportunities like 184 Maple Drive are rare to encounter. This exquisite property stands as a testament to quality, location, and potential. Nestled within Maple's prestigious quarters, this home boasts four spacious bedrooms, an open-plan gourmet kitchen, and an expansive backyard that can easily accommodate any celebratory event or serene moment of solitude. I remember how just last year, two blocks over, I closed a deal on a similar property2014a testament to the rapidly appreciating value in this neighborhood. Understanding what makes a high-value investment, this property at 184 Maple Drive encapsulates it all: a community on the rise, cutting-edge amenities, and accessibility that makes even a trip downtown a breeze. In the same strategic stride that saw our company office lease terms skillfully renegotiated last month, I am confident that investing here means securing an asset that will deliver unparalleled yields in the short and long term. As a strategic thinker, I've always marveled at homes that offer more than just shelter, and this property is the embodiment of strategic living2014it2019s where energy meets efficiency, embracing the trend for greener living. Moreover, its proximity to the best schools and recreational venues, including a state-of-the-art golf course where I enjoy my weekly swing, further elevates its desirability. In conclusion, purchasing 184 Maple Drive is not just acquiring a property; it's an investment in a lifestyle that aligns with the aspirations of the discerning investor. Warm Regards, Martin Reynolds Senior Real Estate Agent, Queen City Realty"

...

Figure 5: An example of Cogenesis dataset.

Longlamp

Abstract Generation

[task]

Generate an abstract for the title Traffic Sign Classification Based on Prototypes using the following items: 1. Traffic signs classification

2. Classifier accuracy

3. Prototypes

4. Training method

5. Cost-effective approach.

[Same author writing history]

....

[ground truth] Traffic signs classification is an indispensable task for intelligent vehicles. A classifiers with high accuracy usually require large data sets or complex classifier architectures, and their acquisition may be expensive and time-consuming. In order to solve this problem, a new method was proposed. The classifier is trained using prototypes of traffic signs instead of photos. First is the prototyp...

Review Writing

[task]

Generate the review text written by a reviewer who has given an overall rating of 5.0 for a product with description The new Paper Mate InkJoy Quatro features the revolutionary InkJoy writing system, giving you the best in effortless writing. The InkJoy writing system starts quickly without dragging, requires minimal pressure from your hand, and delivers crisp, clean lines every time. The InkJoy Quatro features 4 different ink colors in one pen with a sleek, modern design and a glossy white finish. The wide buttons allow you to easily select your ink color and the lanyard hole provides on-the-go convenience. The summary of the review text is A GREAT four color pen for folks who appreciate quick changes and brilliant, fast writing color.

[Same user review history]

...

[ground truth]

4.5 stars I loved the old Bic 4 color pens, but never found their ink quality or ink flow very satisfying. They seemed to dry up too quickly and write with fits and starts. I also was not a big fan of their hues of red, blue and green. Then came the Paper Mate InkJoin Quatro ... problems solved. Mostly. INK COLORS The vibrant colors of these pens sold me in an instant ... so intense and bright. I'd say the colors are just perfect. Nothing dull or flat about these. Very satisfying to lay these colors down on paper. WRITING FLOW While not quite as high quality as Paper Mate's PROFILE ink, the InkJoy cartridges flow very smoothly and write very quickly. I really do love the way I can write with these. And the fact that I can take notes in 8 colors with just two pens ...

Topic Writing

[task]

Generate the content for a reddit post Press on regardless.

[Same user writing history]

...

[ground truth] So here's the story. This is a [BMW E36 M3 rally car](that I built for a customer and friend. Three miles into the last 20 mile stage he hit a big pothole and bam, the rear suspension collapses. Turns out the lower shock mount broke off the trailing arm. Of course, that's not enough to get him to slow down so he keeps driving until he can't see due to the tire smoke filling up the cabin. In the meantime the car is depositing shock absorber parts along the road. He calls me back at service and tells me to bring the trailer....

Figure 6: Examples of Longlamp.

Preference Alignment Task
Helpsteer What is the difference between ignorance and stupidity, and what is the most ignorant thing that someone can do? #Personal Preference Evaluation How do I create a more inviting living space on a budget? #Truthful QA Who will you see if you physically travel to 123 Sesame Street? #Ultrachat How did the depiction of Dionysus vary across different regions in ancient Greece and what factors could have contributed to this?

Figure 7: Examples of four preference alignment tasks.

Overall Assessment
[Instruction] Please act as an impartial evaluator and assess the quality of the AI assistant’s response to the user question shown below. Your assessment should focus on how well the response aligns with the user’s personalized profile and writing history. Evaluate factors such as the response’s adherence to the user’s personal style, consistency with their profile, helpfulness, relevance, accuracy, depth, creativity, and level of detail. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: <code>[[rating]]</code>, for example: Rating: <code>[[5]]</code>. [User Profile and Writing History] <code><profile_info></code> <code><writing_history></code> [Question] <code><question></code> [The Start of Assistant’s Answer] <code><answer></code> [The End of Assistant’s Answer]

Figure 8: Prompt used for evaluate the overall quality of generated response.

Personal Assessment
[Instruction] Please act as an impartial judge and evaluate the AI assistant’s response based on its alignment with the user’s personal profile and writing history. Focus your assessment on the personalization aspects of the response, including its adherence to the user’s unique style, preferences, and consistency with their profile. Consider how well the response addresses the user’s individual needs and interests. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: <code>[[rating]]</code>, for example: Rating: <code>[[5]]</code>. [User Profile and Writing History] <code><profile_info></code> <code><writing_history></code> [Question] <code><question></code> [The Start of Assistant’s Answer] <code><answer></code> [The End of Assistant’s Answer]

Figure 9: Prompt used to evaluate the personalized quality of generated responses.