

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 BREAKING STATIC PARADIGMS: A MUTUAL EVOLU- TION FRAMEWORK FOR EDGE-CLOUD MODEL COL- LABORATION

006 **Anonymous authors**

007 Paper under double-blind review

011 ABSTRACT

013 To simultaneously achieve high performance and low latency, the paradigm of
 014 *edge-cloud* model collaboration, where Large Language Models (LLMs) are
 015 deployed on the powerful cloud and Small Language Models (SLMs) on the
 016 resource-limited edge devices, has garnered great attention recently. However, a
 017 key limitation of current edge-cloud architecture is its static nature, which hinders
 018 the dynamic integration of new knowledge. More specifically, existing methods
 019 typically update the system by directly retraining the cloud-based LLM with edge-
 020 side newly collected data, which not only increases communication overhead but
 021 also neglects available computing power and data accessibility on edge devices.
 022 To tackle this challenge, we propose a novel mutual Evolution framework for
 023 edge-cloud model Collaboration called **CoEvo** that enables both cloud-side LLM
 024 and edge-side SLMs to update with new knowledge continuously. The cloud-
 025 based LLM can enhance edge-side SLMs through credible Chain-of-Thought
 026 (CoT) based knowledge distillation to improve its general understanding capa-
 027 bilities. Once the edge-side SLMs collect new domain-specific knowledge and
 028 optimize themselves locally, they will specifically enhance the cloud-based LLM
 029 via a credible probability matrix predicted on a few samples without uploading
 030 all raw data. Through this mutual evolution, the system can achieve continual
 031 optimization of the cloud and edge-side models and promote real-world deploy-
 032 ments. Experimental results demonstrate a considerable performance gain of our
 033 edge-side SLMs against existing methods on the target dataset, with the cloud-side
 034 LLM also achieving a notable improvement over the base model.

035 1 INTRODUCTION

036 Large language models (LLMs) such as the GPT series (Achiam et al., 2024) and DeepSeek R1
 037 (Guo et al., 2025) have been extensively adopted across diverse domains, substantially improving
 038 operational efficiency and fostering innovation in a wide range of industries (Saha et al., 2025). To
 039 support the practical deployment of these models, the edge–cloud collaborative architecture (Wang
 040 et al., 2024b) has emerged as a critical paradigm. This architecture capitalizes on the abundant
 041 computational resources of cloud servers to maintain large-scale LLMs, while simultaneously deploying
 042 lightweight small language models (SLMs) on resource-constrained edge devices (Tian et al., 2024).
 043 By integrating the powerful inference capabilities of LLMs with the lightweight and efficient deploy-
 044 ment of SLMs (Zhao et al., 2024), the edge–cloud architecture achieves an effective balance between
 045 performance and efficiency. In particular, techniques such as LLM-guided inference enable SLMs
 046 (Liu et al., 2024) to inherit knowledge and enhance their reasoning abilities, all while sustaining
 047 low latency and minimizing computational overhead. This collaborative paradigm thus provides a
 048 promising pathway for scaling LLM applications to real-world environments.

049 In recent years, researchers have made notable progress in improving the inference performance
 050 of edge-cloud architectures. By leveraging maintained external knowledge bases, language models
 051 enhance overall performance on both edge and cloud sides through retrieval-augmented generation
 052 (RAG) (Lewis et al., 2020; Liu et al., 2025; Qin et al., 2025). To avoid substantial external storage
 053 and query overhead during the inference stage, some studies have utilized external tools (Zhuang
 et al., 2023; Yuan et al., 2024a) such as search engines and compilers to assist language models in

054 solving practical problems. (Chen et al., 2024a) utilizes detection algorithms to filter data, retaining
 055 routine data processing on the edge side while uploading only critical data to the cloud, thereby
 056 maintaining the inference performance of the framework and reducing redundant transmission. The
 057 authors in (Yao et al., 2024) maintain a database built on the edge side that stores historical requests
 058 and responses from cloud-based LLMs, effectively enhancing the knowledge richness of SLMs and
 059 improving the response quality for similar requests.

060 Although these works have achieved great progress in the edge-cloud model collaboration, they all
 061 operate in a static mode (Qin et al., 2024), lacking the ability to dynamically acquire and learn new
 062 information, severely restricting their adaptability and real-time learning potential. Some existing
 063 methods leverage edge devices to collect new data and upload it to the cloud, achieving dynamic
 064 knowledge integration through periodic updates of the cloud-based LLM (Fan et al., 2023; Kuang
 065 et al., 2024). However, such an update scheme is overly simple and exposes the entire architecture
 066 to several critical challenges: ~~(1) directly uploading raw data to the cloud for updating LLMs incurs~~
 067 ~~substantial communication overhead~~; (1) edge devices must wait for data transmission and cloud-
 068 side model updates, which leads to latency in user experience; (2) since new data are collected on
 069 edge devices, uploading them to the cloud for centralized updates essentially neglects the computa-
 070 tional capacity of edge devices, reducing resource utilization; and (3) in some cases, the data may
 071 involve sensitive information, making it unsuitable for direct uploading to the cloud for training.

072 Offloading some computational tasks to edge devices presents a highly attractive alternative. Edge
 073 devices, such as the NVIDIA Jetson series, embedded systems with discrete GPUs or onboard in-
 074 telligent computing platforms, are typically equipped with powerful CPUs, GPUs, or dedicated AI
 075 accelerators, possessing sufficient computing power to support lightweight fine-tuning of models
 076 with billions of parameters. This capability is further enhanced by Parameter-Efficient Fine-Tuning
 077 (PEFT) techniques, which significantly reduce the computational overhead required for fine-tuning
 078 with minimal impact on model performance. ~~We compare the data transfer overhead between edge-
 079 based and cloud-based update strategy, as detailed in Table 1. The cloud-based update strategy~~
 080 ~~requires two data transmissions between the edge and cloud to achieve a collaborative update (even~~
 081 ~~with local storage), while the edge-based approach only needs one edge-to-cloud transmission.~~

082 To address these chal-
 083 lenges, we propose an
 084 enhanced edge-cloud
 085 architecture that is empow-
 086 ered with the capability
 087 to efficiently ~~and securely~~
 088 integrate newly collected
 089 knowledge. This improve-
 090 ment not only enables
 091 continuous adaptation of
 092 deployed models to evolving data distributions but also ensures that knowledge updates can be
 093 incorporated with minimal latency ~~and without compromising the privacy of user data~~. In a standard
 094 edge-cloud system that serves users, cloud-based LLMs typically play an assisting and guiding role,
 095 while SLMs deployed on edge devices interact directly with users. During this process, edge device
 096 SLMs can directly access and learn from newly generated real-time data samples, thereby achieving
 097 more accurate modeling of the local data characteristics, which facilitates self-updates with minimal
 098 overhead. Driven by this intuition, we propose a novel mutual Evolution framework for edge-cloud
 099 model Collaboration called **CoEvo** that enables both cloud-side LLM and edge-side SLMs to
 100 continuously update with new knowledge. More specifically, it consists of two independent stages
 101 to update the cloud-side and edge-side models. In the cloud-to-edge stage, CoEvo incorporates
 102 confidence (Xiong et al., 2024) scores into the Chain-of-Thought (CoT), teaching the edge-side
 103 SLMs to generate high-confidence responses that emulate the cloud-based LLM. While in the
 104 edge-to-cloud stage, the edge-side SLMs continuously acquire new domain-specific data and
 105 update themselves. CoEvo then performs credibility-based filtering on the SLMs’ newly learned
 106 representations, allowing only highly reliable domain knowledge to be uploaded and used to
 107 enhance specialized inference capabilities of the cloud-side LLM. The major contributions of this
 108 paper are summarized as follows:

Table 1: Comparison of data transfer overhead between edge-based and cloud-based update strategy.

Method	Commonsense CQA	Math GSM8K	Natural Language Winogrande
Cloud-based update strategy			
w/o Local data storage	2.0 \times	1.6 \times	1.8 \times
Local data storage	1.3 \times	1.2 \times	1.2 \times
Edge-based update strategy			
	1.0\times	1.0\times	1.0\times

- 108 • We are the first to explore mutual evolution in edge-cloud model collaboration, breaking
109 the static paradigm of traditional edge-cloud architectures and enabling an efficient **and**
110 **secure** continual learning of new knowledge.
- 111 • We propose CoEvo, an enhanced edge-cloud architecture that enables perception and learn-
112 ing from raw data through local updates on the edge side. It also facilitates bidirectional
113 knowledge transfer between the edge and the cloud via a credible chain-of-thought and
114 credible probability matrices.
- 115 • We conduct extensive experiments across multiple datasets spanning various domains. Ex-
116 perimental results demonstrate that our method achieves performance improvements on
117 both the edge and cloud sides compared to state-of-the-art approaches.

119 We employ Llama3 (Dubey et al., 2024) 8B as the edge-side SLMs and Llama3 70B as the cloud-
120 side LLM in the edge-cloud architecture, evaluating performance across general domains (MMLU
121 (Hendrycks et al., 2021)), commonsense reasoning tasks (CommonsenseQA(CQA) (Talmor et al.,
122 2019)), math tasks (GSM8K (Cobbe et al., 2021)), and co-reference resolution tasks (WinoGrande
123 (Sakaguchi et al., 2020)). In the cloud-to-edge process, CoEvo improves by 1% to 2% compared to
124 existing baseline methods; In the edge-to-cloud process, CoEvo enables a 1% to 1.3% increase in
125 the inference accuracy of the optimized cloud-side LLM.

126 2 RELATED WORK

127 **Knowledge Transfer from Cloud LLMs to Edge SLMs:** Based on edge-cloud architecture, (Xu
128 et al., 2024; Peng et al., 2024) use cloud-based LLMs to enhance the performance of edge-side
129 SLMs. This is achieved by building a local data store from historical interactions with the cloud
130 LLM and dynamically integrating it with the predictions of the SLM on the device during inference
131 (Ding et al., 2024). (Chen et al., 2024b) focuses on improving edge-side SLMs through knowledge
132 distillation (Wang et al., 2022) while offloading all gradient-related operations to the cloud, thereby
133 reducing the computational burden on the edge-side. (Hao et al., 2024; NING et al., 2025) leverage
134 LLMs to provide token-level inference guidance for edge-side SLMs, integrating the LLM’s semantic
135 understanding into the actual inference process of the SLM. However, it is constrained by the
136 upper limit of the cloud-side model’s inference capabilities, and the lack of ground truth labels and
137 chain of thought(CoT) (Wei et al., 2022b) data for given tasks (Yuan et al., 2024b) undermines its
138 applicability. Our approach focuses on the potential of edge devices to acquire new data and aims to
139 achieve continual learning in an edge-cloud framework by leveraging domain-specific data.

140 **Advanced Inference Techniques in Language Model:** The CoT-related (Wang et al., 2023b;
141 Wan et al., 2025; Zhang et al., 2025a) technique guides language models to generate coherent
142 thought chains and answers during inference, requiring the model to engage in one or more inter-
143 mediate reasoning steps before producing the final answer (Kojima et al., 2022; Fu et al., 2023). Other
144 studies have further enhanced the inference capabilities of language models by extending the
145 CoT paradigm, such as integrating internal generation processes with external actions (Wang et al.,
146 2024a) (e.g., leveraging RAG, invoking search engines, calculators, or code interpreters). They
147 enhance the model’s comprehensive inference capabilities by leveraging external knowledge, although
148 constructing a well-structured external knowledge base or designing effective task flows is by no
149 means an extra overhead (Cheetirala et al., 2025). The issue with inference technology is that lan-
150 guage models themselves are constrained (Bian et al., 2024; Wang et al., 2025) by the scope of
151 training data, delays in knowledge updates, and potential factual biases. Relying solely on inference
152 optimization often struggles to break through the model’s inherent cognitive boundaries.

153 **Fine-Tuning Techniques for Language Model Optimization:** Direct Preference Optimization
154 (DPO) (Rafailov et al., 2023) eliminates the need for reward models typically required in rein-
155 forcement learning and RLHF (Reinforcement Learning from Human Feedback) (Ouyang et al., 2022)
156 by directly incorporating preference data into the training objective. This approach reduces com-
157 putational overhead while ensuring the model focuses on preferred outputs (Shankar et al., 2024).
158 Distill-step-by-step (Hsieh et al., 2023) enhances student models’ inference capabilities by aligning
159 their outputs (both answers and rationales) with those generated by teacher models (Beyer et al.,
160 2022). This dual alignment improves both the accuracy of problem solving and the generation of
161 CoT. Chain of Preference Optimization (CPO) (Zhang et al., 2025b) extends DPO by incorporating
multistep thought chains, where the model generates and evaluates multiple inference paths while

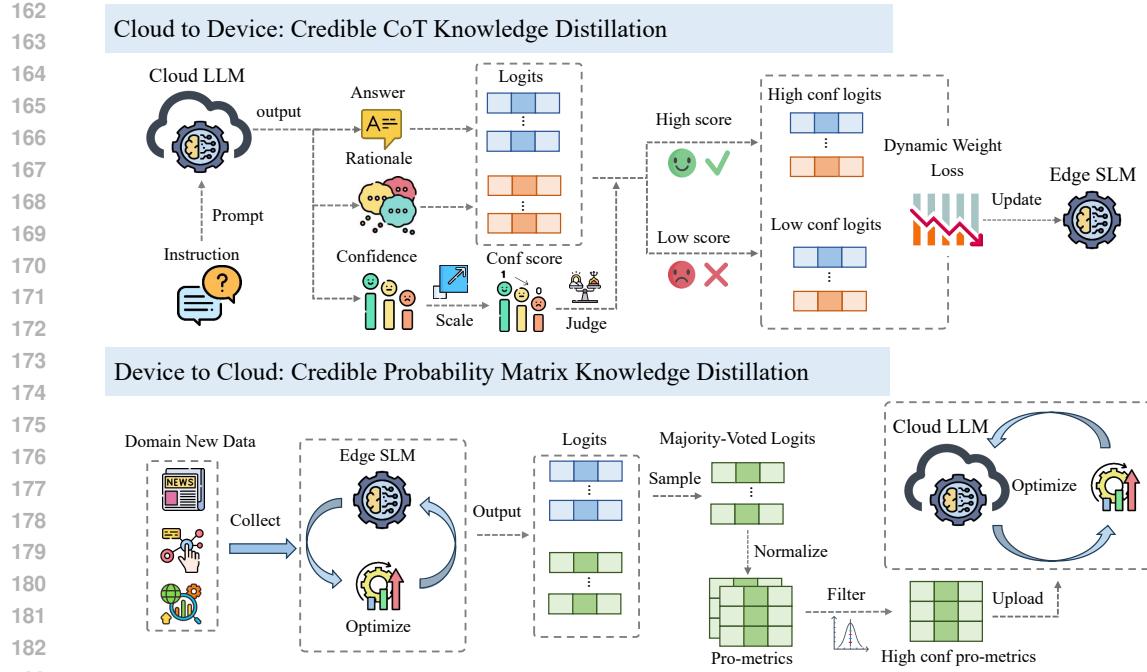


Figure 1: The overall architecture of CoEvo. In the cloud-to-edge phase, the cloud LLM performs instruction-guided inference to generate answers, rationales, and confidence scores. These outputs provide the basis for knowledge distillation, where confidence scores dynamically weight the knowledge to optimize edge SLMs. In the edge-to-cloud phase, the edge-side SLMs are optimized with new data and generate responses from historical interactions. CoEvo then applies multiple-sample voting, probability scaling, and filtering to extract high-quality domain knowledge, which is uploaded back to the cloud to further refine the cloud-side LLM.

explicitly considering dispreferred chains. A potential issue with these methods is the lack of consideration for knowledge quality, which leads to suboptimal results being incorporated into the training process. Our approach filters the data used to avoid interference from low-quality content, thereby enhancing the effectiveness of fine-tuning.

3 COEVO: A MUTUAL EVOLUTION FRAMEWORK FOR EDGE–CLOUD MODEL COLLABORATION

3.1 OVERVIEW

CoEvo leverages the unique advantage of edge devices being accessible to users and extends the static edge–cloud collaboration framework into a dynamic knowledge learning paradigm, supporting the mutual evolution of models on both the edge and cloud sides. Figure 1 illustrates the details of CoEvo. In the cloud-to-edge phase, the cloud-based LLM generates rationales and labels on a general-domain dataset D . These outputs are distilled into a base edge-side SLM \mathcal{M}_0 , where confidence scores are used to weight the knowledge and mitigate the impact of suboptimal outputs, resulting in an enhanced SLM \mathcal{M}_1 with improved semantic comprehension. \mathcal{M}_1 is then deployed on edge devices for domain-specific inference. In the edge-to-cloud phase, the edge-side SLMs continuously interact with the local environment and user context, serving as natural collectors of domain-specific data and enabling efficient local updates. Through this process, \mathcal{M}_1 is further optimized into a domain expert model \mathcal{M}_2 . The superior domain knowledge extracted from \mathcal{M}_2 , specifically knowledge that surpasses the cloud LLM’s existing domain understanding, is selectively distilled back into the cloud-side LLM \mathcal{M}_t , yielding an improved model \mathcal{M}_T with stronger domain inference capabilities. Through this bidirectional synergy, both edge-side and cloud-side models can co-evolve by continuously learning new knowledge and improving domain-specific performance.

216
217

3.2 CLOUD-TO-EDGE: CREDIBLE COT KNOWLEDGE DISTILLATION

218
219
220
221
222
223
224
225
226
227
228
229

In the cloud-to-edge stage of CoEvo, the cloud-based LLM transfers its inference capability to the edge-side SLM through knowledge distillation. Conventional methods typically rely on labels as the primary form of knowledge. However, labels alone are insufficient to enhance the semantic understanding of SLMs, limiting their ability to achieve strong domain-specific inference even after optimization. Motivated by works such as CoT distillation (Wang et al., 2023a), we instead use both labels and rationales as knowledge, thereby expanding the informational scope and enabling SLMs to learn not only outcomes but also the underlying inference processes. A critical challenge lies in the assumption that cloud-based LLMs can consistently generate labels and rationales of consistently high quality. In practice, even LLMs, despite their strong inference capabilities, may produce ambiguous or erroneous outputs, which can degrade performance. To address this issue, CoEvo introduces mechanisms that allow SLMs to emphasize high-quality knowledge while filtering out noisy or unreliable inferences, thereby maximizing the effectiveness of the distillation process.

230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251

Confidence is commonly used to reflect the degree of self-assurance that large models have in their outputs, which may correlate with the correctness of the inference results. We analyze the confidence generated by LLM inference across multiple datasets from different domains, as shown in Figure 2. Two key observations emerge: (1) LLMs generally exhibit high confidence in their responses (consistently above 0.67), regardless of correctness, which aligns with findings in prior studies; and (2) despite this overall tendency toward high confidence, correct answers are still associated with significantly higher confidence than incorrect ones. These observations suggest a potential correlation between confidence levels and the quality of the responses. In particular, the validity of the inference process (i.e., the generated CoT) is strongly tied to the correctness of the final answer: a coherent rationale typically leads to a correct result. Thus, confidence can serve as a useful indicator of both rationale quality and answer reliability. Nonetheless, we also observe cases where LLMs produce correct answers with low confidence, reflecting a lack of self-assurance. In such cases, multiple sampling often increases the likelihood of generating incorrect responses. Therefore, it is advisable to assign lower weights to these low-confidence samples during training, even if their answers are occasionally correct, to prevent the student model from inheriting similar confusion.

252
253
254
255
256

We let the cloud-based LLM \mathcal{M}_t execute inference tasks on a problem set \mathcal{D} . \mathcal{M}_t generates corresponding rationale ($\mathcal{Q} \rightarrow \mathcal{R}$), answers ($\mathcal{Q} \rightarrow \mathcal{A}$), and confidence ($\mathcal{Q} \rightarrow \mathcal{C}$) for each instruction. We perform knowledge distillation on the SLM. Given the preceding sequence $(1, \dots, i-1)$, \mathcal{M}_t generates a prediction for the i^{th} token. Based on the answers and rationales obtained, the fundamental objectives are formulated as follows:

$$\mathcal{L}_A = -\log p([\mathcal{A}_i \mid \mathcal{P}_{Q \rightarrow A}; Q; A_{<i}]; \mathcal{M}_t). \quad (1)$$

$$\mathcal{L}_R = -\log p([\mathcal{R}_i \mid \mathcal{P}_{Q \rightarrow R}; Q; R_{<i}]; \mathcal{M}_t). \quad (2)$$

260
261
262
263
264
265

where p represents probability, \mathcal{A}_i and \mathcal{R}_i depict the i^{th} token in the answer and rationale, $< i$ refers to the sequence of tokens preceding the i^{th} token, $[\cdot]$ denotes the operation that formats the LLM input by assembling question, answer, and rationale into the prompt \mathcal{P}_t . We scale the training objectives in (1) and (2) using the confidence:

$$\mathcal{L} = S_{conf} \cdot [(1 - \alpha)\mathcal{L}_R + \alpha\mathcal{L}_A]. \quad (3)$$

266
267
268
269

where α is hyperparameter, S_{conf} represents the confidence score in the model inference. By modeling knowledge distillation as the process above, we improve SLM \mathcal{M}_0 to obtain \mathcal{M}_1 . \mathcal{M}_1 is deployed on an edge device to perform practical tasks. Not only does \mathcal{M}_1 surpass \mathcal{M}_0 in inference ability, but it also possesses stronger cognitive understanding abilities. This makes it possible for \mathcal{M}_1 to evolve into a domain-specific expert after being exposed to domain-specific data.

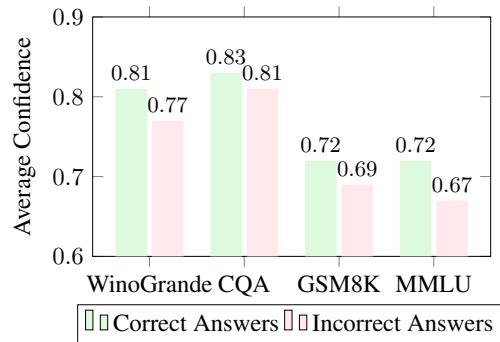


Figure 2: Comparison of average confidence scores for correct vs. incorrect answers across different groups.

270 3.3 EDGE TO CLOUD: CREDIBLE PROBABILITY MATRIX KNOWLEDGE DISTILLATION
271

272 CoEvo primarily leverages edge-side SLMs to acquire new data. By learning from evolving data
273 distributions and performing local self-updates, the SLMs dynamically adapt to new knowledge.
274 These locally acquired updates are then fed back to the cloud, where they further enhance the per-
275 formance of the LLM. We first leverage the newly acquired data to optimize SLM \mathcal{M}_1 on the edge
276 device, resulting in a domain-specific expert \mathcal{M}_2 . While inheriting \mathcal{M}_1 's strong inference abilities,
277 \mathcal{M}_2 also possesses richer domain-specific knowledge. We select a portion of the instructions used
278 during the self-updating process and combine them with \mathcal{M}_2 's responses to these instructions as
279 knowledge to be uploaded to the cloud. To obtain superior responses, for each instruction, CoEvo
280 samples multiple inference results y and selects the one with the highest consistency as the inter-
281 action record y^* by majority voting. Through inference sampling, CoEvo aims to obtain responses
282 that approach the upper limit of \mathcal{M}_2 's inference capability.

$$283 y^* = \arg \max_{y \in \mathcal{Y}} \sum_{i=1}^N \mathbb{I}(y_i = y). \quad (4)$$

285 We selectively extract interaction records in order to identify and upload the knowledge where \mathcal{M}_2
286 produced relatively high-quality responses to the cloud **while minimizing the impact of transmitting**
287 **raw responses on original data privacy**. For each response of \mathcal{M}_2 , we obtain the probability matrix
288 P of its response component:

$$289 \mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nm} \end{pmatrix}, \quad p_{ij} = \frac{e^{z_{ij}}}{\sum_{k=1}^m e^{z_{ik}}}. \quad (5)$$

294 where m is the length of the response sequence, z represents the logits corresponding to the i^{th}
295 token in the response sequence, and n is the size of the vocabulary. **Due to the large value of n (often**
296 **tens or hundreds of thousands), computing and transmitting the probability matrix for the entire**
297 **sequence becomes computationally expensive.** To address this, CoEvo therefore applies a compression
298 process. Through statistical analysis across the multiple datasets used in our experiments, we find
299 **that the probability distribution is highly concentrated in the top-k elements.** We select the top 10
300 **elements, which significantly reduces the raw transmission cost while striving to preserve the uncer-**
301 **tainty information inherent in the model's predictions.** To obtain superior responses, we reconsider
302 the relationship between confidence and inference outcomes. Confidence in a language model's re-
303 sponse is directly reflected in the magnitude of its larger output logits. This same principle applies
304 to evaluating the reliability of a probability matrix. For a given interaction record with a probability
305 matrix, CoEvo calculates: (1) the sum of the top-k probabilities and (2) the maximum probability in
306 the top-k probabilities.

$$306 p_{t1} = \begin{cases} 1, & \text{if } \sum_{i=1}^k P_{ij} > p_1 \quad \forall j \in \{1, \dots, m\}, \\ 0, & \text{otherwise.} \end{cases}, \quad p_{t2} = \begin{cases} 1, & \text{if } P_{1j} > p_2 \quad \forall j \in \{1, \dots, m\}, \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

309 If $p_{t1}=1$ and $p_{t2}=1$, the interaction record is considered high quality:

$$311 p_{final} = \begin{cases} 1, & \text{if } p_{t1} = 1 \wedge p_{t2} = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

313 We upload records with $p_{final}=1$ to the cloud, feeding high-quality domain knowledge back to the
314 cloud-side LLM. Through credible probability matrix knowledge distillation, we optimize the LLM
315 \mathcal{M}_t to obtain \mathcal{M}_T . \mathcal{M}_T breaks through the original performance ceiling of the cloud-side LLM
316 by achieving dynamic updates based on edge-side new knowledge and serves as the foundational
317 cloud-side LLM for the next iteration of CoEvo.

318
319 4 EXPERIMENTS
320321 4.1 EXPERIMENT SETUP
322

323 **Datasets.** We evaluate CoEvo on four datasets from diverse domains: (1) Multi-task understanding:
MMLU (Hendrycks et al., 2021); (2) Commonsense inference: CQA (Talmor et al., 2019); (3)

324
325

Table 2: Performance comparison between CoEvo and baselines in the cloud-to-edge stage

326
327
328
329

Method	Multi-task MMLU	Commonsense CQA	Math GSM8K	Natural Language WinoGrande
Meta Llama3 8B	57.26	59.38	52.15	62.46
Efficient inference based on CoT				
Chain of Thought	59.48	63.68	59.22	62.77
Self-Consistency CoT	61.17	65.00	60.09	62.98
Tree of Thought	59.95	64.64	58.73	62.48
Fine-tuning-based optimization				
DPO	63.90	64.06	61.66	67.10
Distill step by step	64.38	65.44	59.31	68.20
SPIN	63.85	63.19	62.08	66.62
CoEvo (Ours)	65.91	66.26	62.44	70.40

340

341
342
343

Math problems: GSM8K (Cobbe et al., 2021); (4) Natural Language Inference: **WinoGrande** (Sakaguchi et al., 2020). Details of the datasets can be found in Appendix A.

344
345
346
347
348
349
350
351

Baseline. We separately evaluate the cloud-to-edge and edge-to-cloud processes to validate the performance improvements of both the edge-side SLM and cloud-side LLM. For the cloud-to-edge process, we compare two types of approaches. (1) Efficient inference methods based on CoT: **Chain of Thought** (Wei et al., 2022b), **Self-Consistency CoT** (Wang et al., 2023b), and **Tree of Thought** (Yao et al., 2023); (2) Fine-tuning-based optimization methods: **DPO** (Rafailov et al., 2023), **Distill Step-by-Step** (Hsieh et al., 2023), and **SPIN** (Chen et al., 2024c). For the edge-to-cloud process, we compare the results with both the base LLM and the LLM after Supervised Fine-Tuning (SFT) (Wei et al., 2022a). Details of the baselines can be found in Appendix B.

352
353
354
355
356
357
358
359

Implementation Details. We use Llama3-8B (Dubey et al., 2024) as the edge-side SLM and Llama3-70B as the cloud-side LLM in our experiments. **On the edge side, we limit our experiments to 1 A100 80GB GPU (under 42 GB VRAM used in total) to simulate limited computational resources; On the cloud side, we employ a batch size of 32 or 64 using 16 A100 80GB GPUs.** The training phase employs the AdamW optimizer with cosine annealing and 20 warmup steps. We employ machine learning libraries such as Deepspeed (Rajbhandari et al., 2020) to facilitate model training **and utilize the LoRA technique to reduce the resource overhead required for edge-based training.** Details of configurations can be found in Appendix C.

360
361

4.2 PERFORMANCE OVERVIEW

362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377

Table 2 presents a comparison between CoEvo and baseline methods in cloud-to-edge stage. CoEvo consistently outperforms most baseline methods across the four task domains. On MMLU, CoEvo achieves a gain of 1.5% over the strongest baseline, demonstrating its effectiveness in cross-domain tasks. On CQA, it improves by 0.8% compared to the best baseline, achieving a marginal gain. Our analysis reveals that the answers generated by cloud-side LLM in cloud-to-edge stage were actually suboptimal, since the use of CoT-based inference on CQA reduced the accuracy of the inference. This may be attributed to the commonsense nature of CQA, for which the CoT method is less suitable. On WinoGrande, CoEvo outperforms the best baseline by approximately 2%, demonstrating the effectiveness of our method in enhancing the SLM’s semantic understanding ability. On GSM8K, CoEvo exhibits slightly **higher** performance compared to the best baseline. **A potential explanation is the fact that in the mathematical domain, the evaluation of CoT quality is not fixed. For example, multiple solution approaches can lead to correct results, and this characteristic may have interfered with CoEvo’s confidence-based evaluation process.** Fur-

Table 3: Performance comparison in edge-to-cloud stage.

Method	Commonsense CQA	Math GSM8K	Natural Language WinoGrande
Meta Llama3 70B	78.09	80.20	82.42
SFT	79.09	81.74	83.97
CoEvo(Ours)	79.26	81.18	83.79

Table 4: Performance analysis of different design choices.

Ablation	Confidence	Inference	Response	Commonsense	Math	Natural Language
ID	Evaluation	Sampling	Filtering	CQA	GSM8K	WinoGrande
1	Yes	No	No	78.64	79.19	81.13
2	No	Yes	No	78.71	79.41	81.34
3	No	No	Yes	78.02	79.01	81.47
4	Yes	Yes	No	78.71	80.76	83.72
5	Yes	No	Yes	78.84	80.44	83.50
CoEvo	Yes	Yes	Yes	79.26	81.18	83.79

thermore, while CoT-based baseline methods lead to a multiplicative increase in inference latency, CoEvo requires only single step inference to generate responses, thereby preserving the low-latency requirement of the edge-side SLM in edge-cloud architecture. In contrast to methods like DPO that require the parallel training of both a base and a policy model, CoEvo’s single-model training mode in cloud-to-edge stage significantly reduces computational overhead and simplifies the training pipeline. More discussions and results on model performance and communication efficiency are available in Appendix D.

Table 3 compares the performance of CoEvo in the edge-to-cloud stage with the base model and the SFT-tuned LLM. CoEvo achieves a gain of 1.1% to 1.3% over the base model on CQA and WinoGrande, demonstrating its effectiveness in leveraging SLM to inversely optimize the LLM within the edge-to-cloud framework. On GSM8K, CoEvo improves by nearly 1% compared to the original model, indicating that CoEvo remains effective for mathematical tasks. Moreover, the performance of CoEvo is comparable to that of the LLM optimized with SFT on all datasets. This demonstrates the robust adaptability of the self-updating SLM at the edge to novel data distributions and the high quality of the knowledge uploaded to the cloud. CoEvo enables edge-to-cloud model feedback optimization in most domains and exhibits promising potential for continual learning.

4.3 ABLATION STUDY

We analyze the effectiveness of each subdesign in CoEvo in different stages, as shown in Table 4. We evaluate the following three settings: (1) Whether to use confidence in cloud-to-edge stage; (2) Whether to perform multiple sampling during response generation in edge-to-cloud stage; and (3) Whether to filter candidate responses in edge-to-cloud stage. The results indicate that incorporating confidence consistently outperforms using only label and rationale data. Furthermore, performing multiple sampling during response generation in the edge-to-cloud stage can enhance the diversity and quality of output. Meanwhile, filtering candidate responses generally leads to better performance than using all responses. These findings suggest that every component of CoEvo is indispensable and effective, collectively contributing to its robust performance across diverse scenarios.

Furthermore, we conduct a detailed evaluation of each component of CoEvo to explore potential optimal configurations. Specifically, we examine: (1) the impact of different confidence weight coefficients on the optimization of edge-side SLMs; (2) the effect of varying sampling counts on the feedback performance of the cloud-side LLM; and (3) the influence of different filtering strategies on the cloud feedback process. The results are presented in Figure 3. We observe that reducing the confidence weight degrades performance, confirming the effectiveness of CoEvo confidence acquisition and scaling strategy. Increasing the sampling count generally improves inference performance, though with diminishing marginal returns, which highlights the need to balance computational cost against performance gains. Finally, applying p_{t1} or p_{t2} individually for data filtering yields consistently inferior results compared to their joint use. This validates CoEvo’s dual-threshold filtering strategy, which considers both the highest and relatively high confidence values, thereby enhancing robustness beyond what a single-threshold approach can achieve.

As mentioned in Section 3.3, during the edge-to-cloud phase, CoEvo utilizes a credible probability matrix as the primary knowledge to optimize the LLM, which differs from the commonly used label-based fine-tuning strategies. We analyze the impact of CoEvo, conventional methods, and

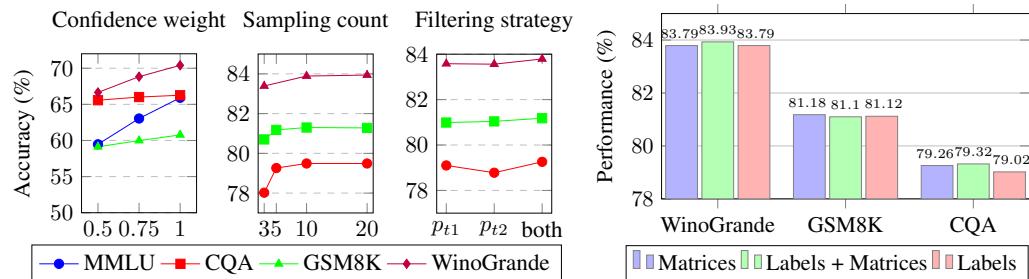


Figure 3: Additional ablation studies on the confidence weight (cloud-to-edge stage), sampling count, and filtering strategy (edge-to-cloud stage).

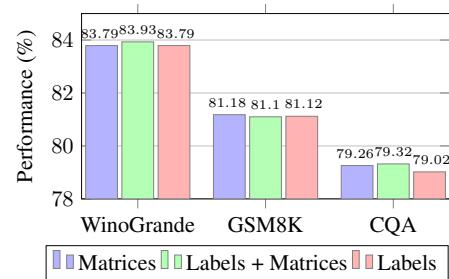


Figure 4: Performance comparison of optimization variants (matrices, labels, and hybrid) in edge-to-cloud stage.

Table 5: Evaluation between SFT-based Model and CoEvo in the edge-to-cloud stage.

Methods	MMLU	CQA	GSM8K	WinoGrande
Meta Llama3 70B	73.24	78.09	80.20	82.42
SFT-based				
\mathcal{M}_{-mmlu}	75.29	78.02	80.18	83.39
\mathcal{M}_{-cqa}	72.08	79.09	80.22	83.43
\mathcal{M}_{-GSM8K}	73.47	77.97	81.74	81.32
\mathcal{M}_{-wino}	72.44	78.08	79.48	83.97
CoEvo				
\mathcal{M}_{T-cqa}	73.18	79.26	80.33	83.09
$\mathcal{M}_{T-GSM8K}$	74.15	78.00	81.18	82.11
\mathcal{M}_{T-wino}	72.89	78.14	79.80	83.79

their combinations on the LLM’s inference performance. Specific results are shown in Figure 4. We observe that using labels, or even a combination of labels and matrices, yields almost comparable or even slightly inferior optimization results for the cloud-based LLM compared to using matrices only. CoEvo, in contrast, utilizes credible probability matrices as knowledge to optimize the cloud-based LLM. This approach excludes the interference from suboptimal outcomes, thereby ensuring both the effectiveness and efficiency of the LLM update process.

4.4 WHY NOT UPLOAD DOMAIN-SPECIFIC DATA DIRECTLY TO CLOUD?

As mentioned in Section 3.3, edge devices can upload new data directly to the cloud to improve LLM. However, the issue is that [directly uploading raw data contradicts the sensitivity to data privacy inherent in the edge-cloud distributed architecture. CoEvo uses response content based on a probability matrix, which partially obscures data details. Furthermore](#), directly fine-tuning the LLM with in-domain data may compromise its original “generalist” nature. As a continually learning edge-cloud architecture, CoEvo must ensure that the general performance of the cloud-side LLM remains consistently high. By using cleaned and filtered probability matrices as fine-tuning data, CoEvo indirectly provides high-quality domain new data while mitigating catastrophic forgetting and minimizing the impact on the LLM’s fundamental inference capabilities. We evaluate the inference performance of the optimized LLM obtained by different methods in other domains.

As shown in Table 5, although the domain-specific inference capabilities of the LLM fine-tuned via SFT on MMLU remain largely unaffected (\mathcal{M}_{-mmlu}), LLMs fine-tuned on other domain-specific datasets exhibit varying degrees of degradation in cross-domain inference performance. Leveraging the knowledge matrix from the feedback of SLM instead of hard labels, CoEvo reduces the interference caused by single domain data on the overall inference ability of the LLM while ensuring stable performance improvement in the target domain.

486
487
488 Table 6: Comparison of computational overhead between CoEvo and simpler strategy.
489
490
491
492
493
494
495

Method	Local optimization	Multiple-sample voting	Probability matrices filtering	Total
transmitting data only	0	0	0	0
update + transmitting data	3.9h	5mins	0	3.98h
CoEvo	3.9h	26mins	1s	4.32h

496 Another factor is the combined cost of computational resources and data transmission in the edge-
497 cloud paradigm. Compared to directly uploading data to the cloud, the local optimization and major-
498 ity voting mechanisms performed on the edge side in CoEvo introduce a certain amount of compu-
499 tational overhead. We analyze the computational overhead of CoEvo in comparison to both directly
500 uploading data to the cloud (transmitting data only) and the method of transmitting data after local
501 updates (update + transmitting data). The specific results are shown in Table 6.

502 "Update + transmitting data" corresponds to the sample with Ablation ID 1 in Table 4. As can
503 be seen, it not only fails to significantly reduce the computational overhead on the edge side but
504 also considerably degrades the performance of cloud-side optimization. Regarding "transmitting
505 data only", this essentially represents the traditional cloud-based update approach, which means
506 it necessitates the additional transmission of the cloud-based LLM's response data to update the
507 edge-side SLM. Detailed results are shown in Table 1.

508 We compare the data transfer overhead between CoEvo and cloud-based update strategy, as detailed
509 in Table 1. The cloud-based update strategy requires two data transmissions between the edge and
510 cloud to achieve a collaborative update. Even with local storage of new data, it still necessitates the
511 additional transmission of the cloud-based LLM's response data to update the edge-side SLM. In
512 contrast, CoEvo completes the SLM update directly on the edge side. It only needs to transmit a
513 portion of the new data and its own responses to the cloud to complete the update of models on both
514 sides, thereby saving the additional overhead of data transmission.

515
516

5 CONCLUSION

517 We present CoEvo, a trainable framework designed to break static mode of edge-cloud architecture
518 and enhance the performance of language models by applying credible CoT knowledge distillation
519 and credible probability matrix knowledge distillation. CoEvo demonstrates the characteristic of
520 proximity to edge devices and users, leveraging the ability to access new domain-specific data dur-
521 ing interaction to enable dynamic knowledge flow in the edge-cloud architecture. Through cloud-to-
522 edge model optimization, CoEvo produces a strengthened SLM with robust inference capabilities;
523 through edge-to-cloud feedback, it yields an LLM with improved performance in specific domains.
524 Furthermore, CoEvo achieves continual evolution through multiple self-iterations, extending its ben-
525 efits across more domains.

526
527

REFERENCES

529 Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Ale-
530 man, and et al. Gpt-4 technical report, 2024. URL <https://arxiv.org/abs/2303.08774>.

532 Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander
533 Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *2022 IEEE/CVF
534 Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10915–10924, 2022. doi:
535 10.1109/CVPR52688.2022.01065.

537 Ning Bian, Xianpei Han, Le Sun, Hongyu Lin, Yaojie Lu, Ben He, Shanshan Jiang, and Bin
538 Dong. Chatgpt is a knowledgeable but inexperienced solver: An investigation of common-
539 sense problem in large language models. In *LREC/COLING*, pp. 3098–3110, 2024. URL
<https://aclanthology.org/2024.lrec-main.276>.

540 Satya Narayana Cheetirala, Ganesh Raut, Dhavalkumar Patel, Fabio Santana, Robert Freeman,
 541 Matthew A Levin, Girish N. Nadkarni, Omar Dawkins, Reba Miller, Randolph M. Steinhagen,
 542 Eyal Klang, and Prem Timsina. Less context, same performance: A rag framework for resource-
 543 efficient llm-based clinical nlp, 2025. URL <https://arxiv.org/abs/2505.20320>.

544 Jiao Chen, Suyan Dai, Fangfang Chen, Zuohong Lv, Jianhua Tang, and Laiquan Han. Edge-cloud
 545 collaborative motion planning for autonomous driving with large language models. In *2024 IEEE*
 546 *24th International Conference on Communication Technology (ICCT)*, pp. 185–190, 2024a. doi:
 547 10.1109/ICCT62411.2024.10946488.

548 Yaofu Chen, Shuaicheng Niu, Yaowei Wang, Shoukai Xu, Hengjie Song, and Mingkui Tan. Towards
 549 robust and efficient cloud-edge elastic model adaptation via selective entropy distillation. In
 550 *The Twelfth International Conference on Learning Representations*, 2024b. URL <https://openreview.net/forum?id=vePdNU3u6n>.

551 Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning con-
 552 vertsweak language models to strong language models. In *Proceedings of the 41st International*
 553 *Conference on Machine Learning*, ICML’24. JMLR.org, 2024c.

554 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
 555 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John
 556 Schulman. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.

557 Yucheng Ding, Chaoyue Niu, Fan Wu, Shaojie Tang, Chengfei Lyu, and Guihai Chen. Enhancing
 558 on-device llm inference with historical cloud-based llm interactions. In *Proceedings of the 30th*
 559 *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD ’24, pp. 597–608,
 560 New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi:
 561 10.1145/3637528.3671679. URL <https://doi.org/10.1145/3637528.3671679>.

562 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha
 563 Letman, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. URL <https://doi.org/10.48550/arXiv.2407.21783>.

564 Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang Yang. Fate-
 565 llm: A industrial grade federated learning framework for large language models, 2023. URL
 566 <https://arxiv.org/abs/2310.10049>.

567 Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting
 568 for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*,
 569 2023. URL <https://openreview.net/forum?id=yflicZHC-19>.

570 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, and et al.
 571 Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL
 572 <https://arxiv.org/abs/2501.12948>.

573 Zixu Hao, Huiqiang Jiang, Shiqi Jiang, Ju Ren, and Ting Cao. Hybrid slm and llm for edge-
 574 cloud collaborative inference. In *Proceedings of the Workshop on Edge and Mobile Foun-
 575 dation Models*, EdgeFM ’24, pp. 36–41, New York, NY, USA, 2024. Association for Com-
 576 puting Machinery. ISBN 9798400706639. doi: 10.1145/3662006.3662067. URL <https://doi.org/10.1145/3662006.3662067>.

577 Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Ja-
 578 cob Steinhardt. Measuring massive multitask language understanding. In *International Confer-
 579 ence on Learning Representations*, 2021. URL <https://openreview.net/forum?id=d7KBjmI3GmQ>.

580 Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner,
 581 Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger
 582 language models with less training data and smaller model sizes. In Anna Rogers, Jordan Boyd-
 583 Gruber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics:
 584 ACL 2023*, pp. 8003–8017, Toronto, Canada, July 2023. Association for Computational Linguis-
 585 tics. doi: 10.18653/v1/2023.findings-acl.507. URL <https://aclanthology.org/2023.findings-acl.507/>.

594 Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large
 595 language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on*
 596 *Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates
 597 Inc. ISBN 9781713871088.

598 Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie,
 599 Yaliang Li, Bolin Ding, and Jingren Zhou. Federatedscope-llm: A comprehensive package for
 600 fine-tuning large language models in federated learning. In *Proceedings of the 30th ACM SIGKDD*
 601 *Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 5260–5271, New York,
 602 NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/
 603 3637528.3671573. URL <https://doi.org/10.1145/3637528.3671573>.

604 Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman
 605 Goyal, Heinrich Kütter, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel,
 606 and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In
 607 H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neu-*
 608 *ral Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc.,
 609 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

612 Jingyu Liu, Linjiaen, and Yong Liu. How much can RAG help the reasoning of LLM?, 2025. URL
 613 <https://openreview.net/forum?id=Q6M7bZI09t>.

614 Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang
 615 Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas
 616 Chandra. MobileLLM: Optimizing sub-billion parameter language models for on-device use
 617 cases. In *Forty-first International Conference on Machine Learning*, 2024. URL <https://openreview.net/forum?id=ElGbXbxcUQ>.

618 JIAHONG NING, Ce ZHENG, and Tingting Yang. DSSD: Efficient edge-device deployment
 619 and collaborative inference via distributed split speculative decoding. In *ICML 2025 Workshop*
 620 *on Machine Learning for Wireless Communication and Networks (ML4Wireless)*, 2025. URL
 621 <https://openreview.net/forum?id=5vkXfhUmnn>.

622 Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong
 623 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kel-
 624 ton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike,
 625 and Ryan Lowe. Training language models to follow instructions with human feedback. In *Pro-*
 626 *ceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS
 627 '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

628 Dan Peng, Zhihui Fu, and Jun Wang. PocketLLM: Enabling on-device fine-tuning for person-
 629 alized LLMs. In Ivan Habernal, Sepideh Ghanavati, Abhilasha Ravichander, Vijayanta Jain,
 630 Patricia Thaine, Timour Igamberdiev, Niloofar Miresghallah, and Oluwaseyi Feyisetan (eds.),
 631 *Proceedings of the Fifth Workshop on Privacy in Natural Language Processing*, pp. 91–96,
 632 Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.privatenlp-1.10/>.

633 Ruiyang Qin, Zheyu Yan, Dewen Zeng, Zhenge Jia, Dancheng Liu, Jianbo Liu, Ahmed Abbasi, Zhi
 634 Zheng, Ningyuan Cao, Kai Ni, Jinjun Xiong, and Yiyu Shi. Robust implementation of retrieval-
 635 augmented generation on edge-based computing-in-memory architectures. In *Proceedings of the*
 636 *43rd IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '24, New York,
 637 NY, USA, 2025. Association for Computing Machinery. ISBN 9798400710773. doi: 10.1145/
 638 3676536.3676674. URL <https://doi.org/10.1145/3676536.3676674>.

639 Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Xuanhe
 640 Zhou, Yufei Huang, Chaojun Xiao, Chi Han, Yi Ren Fung, Yusheng Su, Huadong Wang, Cheng
 641 Qian, Runchu Tian, Kunlun Zhu, Shihao Liang, Xingyu Shen, Bokai Xu, Zhen Zhang, Yining
 642 Ye, Bowen Li, Ziwei Tang, Jing Yi, Yuzhang Zhu, Zhenning Dai, Lan Yan, Xin Cong, Yaxi Lu,
 643 Weilin Zhao, Yuxiang Huang, Junxi Yan, Xu Han, Xian Sun, Dahai Li, Jason Phang, Cheng
 644 Yang, Tongshuang Wu, Heng Ji, Guoliang Li, Zhiyuan Liu, and Maosong Sun. Tool learning

648 with foundation models. *ACM Comput. Surv.*, 57(4), December 2024. ISSN 0360-0300. doi:
 649 10.1145/3704435. URL <https://doi.org/10.1145/3704435>.

650

651 Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea
 652 Finn. Direct preference optimization: Your language model is secretly a reward model. In
 653 *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=HPuSIXJaa9>.

654

655 Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations
 656 toward training trillion parameter models. In *SC20: International Conference for High Perfor-*
 657 *mance Computing, Networking, Storage and Analysis*, pp. 1–16, 2020. doi: 10.1109/SC41405.
 658 2020.00024.

659

660 Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason E Weston, and Tianlu Wang. Learning
 661 to plan & reason for evaluation with thinking-LLM-as-a-judge. In *Forty-second International*
 662 *Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=PNRznmmWP7>.

663

664 Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An ad-
 665 versarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artifi-*
 666 *cial Intelligence*, 34(05):8732–8740, Apr. 2020. doi: 10.1609/aaai.v34i05.6399. URL <https://ojs.aaai.org/index.php/AAAI/article/view/6399>.

667

668 Shreya Shankar, J.D. Zamfirescu-Pereira, Bjoern Hartmann, Aditya Parameswaran, and Ian Arawjo.
 669 Who validates the validators? aligning llm-assisted evaluation of llm outputs with human pref-
 670 erences. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and*
 671 *Technology*, UIST ’24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN
 672 9798400706288. doi: 10.1145/3654777.3676450. URL <https://doi.org/10.1145/3654777.3676450>.

673

674 Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A ques-
 675 tion answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and
 676 Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of*
 677 *the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long*
 678 *and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Com-
 679 *putational Linguistics*. doi: 10.18653/v1/N19-1421. URL <https://aclanthology.org/N19-1421>.

680

681 Yuqing Tian, Zhaoyang Zhang, Yuzhi Yang, Zirui Chen, Zhaohui Yang, Richeng Jin, Tony Q. S.
 682 Quek, and Kai-Kit Wong. An edge-cloud collaboration framework for generative ai service pro-
 683 vision with synergetic big cloud model and small edge models. *IEEE Network*, 38(5):37–46,
 684 2024. doi: 10.1109/MNET.2024.3420755.

685

686 Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. Reasoning aware self-consistency: Leverag-
 687 ing reasoning paths for efficient LLM sampling. In Luis Chiruzzo, Alan Ritter, and Lu Wang
 688 (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the*
 689 *Association for Computational Linguistics: Human Language Technologies (Volume 1: Long*
 690 *Papers)*, pp. 3613–3635, Albuquerque, New Mexico, April 2025. Association for Com-
 691 *putational Linguistics*. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.nacl-long.184. URL
 692 <https://aclanthology.org/2025.nacl-long.184>.

693

694 Chaofei Wang, Qisen Yang, Rui Huang, Shiji Song, and Gao Huang. Efficient knowledge dis-
 695 tillation from model checkpoints. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and
 696 Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL
 697 <https://openreview.net/forum?id=01tDq6SjrfW>.

698

699 Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. Scott: Self-
 700 consistent chain-of-thought distillation. *CoRR*, abs/2305.01879, 2023a. URL <https://doi.org/10.48550/arXiv.2305.01879>.

701

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Exe-
 702 cutable code actions elicit better LLM agents. In *Forty-first International Conference on Machine*
 703 *Learning*, 2024a. URL <https://openreview.net/forum?id=jJ9BoXAfFa>.

702 Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha
 703 Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language
 704 models. In *The Eleventh International Conference on Learning Representations*, 2023b. URL
 705 <https://openreview.net/forum?id=1PL1NIMMrw>.

706 Yingchao Wang, Chen Yang, Shulin Lan, Liehuang Zhu, and Yan Zhang. End-edge-cloud collabora-
 707 tive computing for deep learning: A comprehensive survey. *IEEE Communications Surveys
 708 Tutorials*, 26(4):2647–2683, 2024b. doi: 10.1109/COMST.2024.3393230.

709 Zhengxiang Wang, Jordan Kodner, and Owen Rambow. Exploring limitations of LLM capabilities
 710 with multi-problem evaluation. In Aleksandr Drozd, João Sedoc, Shabnam Tafreshi, Arjun Akula,
 711 and Raphael Shu (eds.), *The Sixth Workshop on Insights from Negative Results in NLP*, pp. 121–
 712 140, Albuquerque, New Mexico, May 2025. Association for Computational Linguistics. ISBN
 713 979-8-89176-240-4. doi: 10.18653/v1/2025.insights-1.12. URL <https://aclanthology.org/2025.insights-1.12/>.

714 Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du,
 715 Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *Inter-
 716 national Conference on Learning Representations*, 2022a. URL <https://openreview.net/forum?id=gEZrGCozdqR>.

717 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, and
 718 et al. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo,
 719 S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neu-
 720 ral Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc.,
 721 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

722 Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. Can
 723 LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs.
 724 In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=gjeQKFxFpZ>.

725 Daliang Xu, Wangsong Yin, Hao Zhang, Xin Jin, Ying Zhang, Shiyun Wei, Mengwei Xu, and
 726 Xuanzhe Liu. Edgellm: Fast on-device llm inference with speculative decoding. *IEEE Trans-
 727 actions on Mobile Computing*, 24(4):3256–3273, December 2024. ISSN 1536-1233. doi:
 728 10.1109/TMC.2024.3513457. URL <https://doi.org/10.1109/TMC.2024.3513457>.

729 Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik
 730 Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In
 731 A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in
 732 Neural Information Processing Systems*, volume 36, pp. 11809–11822. Curran Associates, Inc.,
 733 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf.

734 Zhi Yao, Zhiqing Tang, Jiong Lou, Ping Shen, and Weijia Jia. Velo: A vector database-assisted
 735 cloud-edge collaborative llm qos optimization framework. In *2024 IEEE International Confer-
 736 ence on Web Services (ICWS)*, pp. 865–876, 2024. doi: 10.1109/ICWS62655.2024.00105.

737 Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Yongliang Shen, Kan Ren, Dongsheng Li,
 738 and Deqing Yang. EASYTOOL: Enhancing LLM-based agents with concise tool instruction.
 739 In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024a. URL <https://openreview.net/forum?id=3TuG3S68bb>.

740 Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu,
 741 and Jason Weston. Self-rewarding language models. In *Proceedings of the 41st Interna-
 742 tional Conference on Machine Learning*, ICML’24. JMLR.org, 2024b.

743 Jinghan Zhang, Xiting Wang, Weijieying Ren, Lu Jiang, Dongjie Wang, and Kunpeng Liu. Ratt: A
 744 thought structure for coherent and correct llm reasoning. *Proceedings of the AAAI Conference on
 745 Artificial Intelligence*, 39(25):26733–26741, Apr. 2025a. doi: 10.1609/aaai.v39i25.34876. URL
 746 <https://ojs.aaai.org/index.php/AAAI/article/view/34876>.

756 Xuan Zhang, Chao Dut, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference opti-
757 mization: improving chain-of-thought reasoning in llms. In *Proceedings of the 38th International*
758 *Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2025b.
759 Curran Associates Inc. ISBN 9798331314385.

760 Wentao Zhao, Wenpeng Jing, Zhaoming Lu, and Xiangming Wen. Edge and terminal cooper-
761 ation enabled llm deployment optimization in wireless network. In *2024 IEEE/CIC Interna-*
762 *tional Conference on Communications in China (ICCC Workshops)*, pp. 220–225, 2024. doi:
763 10.1109/ICCCWorkshops62562.2024.10693742.

764 Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. Toolqa: A dataset
765 for llm question answering with external tools. In A. Oh, T. Naumann, A. Globor-
766 son, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Pro-*
767 *cessing Systems*, volume 36, pp. 50117–50143. Curran Associates, Inc., 2023. URL
768 https://proceedings.neurips.cc/paper_files/paper/2023/file/9cb2a7495900f8b602cb10159246a016-Paper-Datasets_and_Benchmarks.pdf.

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810 APPENDIX
811812 THE USE OF LARGE LANGUAGE MODELS (LLMs)
813814 In preparing this work, we have used Large Language Models (LLMs) exclusively for the purposes
815 of translation and language polishing. The content, arguments, and conclusions presented herein
816 are entirely my own, and the use of LLMs did not contribute to the generation of original ideas or
817 substantive content.
818819 ETHICS STATEMENT
820821 This work complies with the ICLR Code of Ethics. No human subjects, personally identifiable data,
822 or sensitive datasets were involved. The use of Large Language Models (LLMs) was strictly limited
823 to translation and language polishing; they did not contribute to the generation of original ideas,
824 methodology, results, or conclusions.
825826 REPRODUCIBILITY STATEMENT
827828 We have taken steps to ensure the reproducibility of our work. The experimental setup, includ-
829 ing datasets, preprocessing steps, and hyperparameters, is detailed in Section 4 and the Appendix.
830 Source code and instructions for reproducing our results are available if needed.
831832 A DATASETS
833834 **MMLU:** A large-scale, widely used multi-task dataset that covers 57 distinct subjects, designed to
835 evaluate a model’s breadth of knowledge and inference abilities across various disciplinary domains.
836 It consists of a development set, a validation set, and a test set, comprising approximately 14,000
837 questions in total.
838839 **CommonsenseQA(CQA):** A dataset designed to evaluate models’ commonsense inference capa-
840 bilities. After partitioning, it contains 9,741 training instances and 1,221 test instances.
841842 **GSM8K:** A dataset specifically designed to evaluate models’ mathematical inference capabilities,
843 consisting of grade-school math word problems that require multiple logical steps to solve. It in-
844 cludes a training set of 7,473 samples and a test set of 1,319 samples.
845846 **WinoGrande:** A large-scale natural language inference dataset, primarily used to evaluate the abil-
847 ity to resolve coreference resolution tasks. It consists of a training set with 40,398 samples, a vali-
848 dation set with 1,267 samples, and a test set with 1,767 samples.
849848 Table 7 in the Appendix shows sample examples from each dataset.
849850 B BASELINES
851852 Efficient inference based on CoT:
853854

- **CoT:** It is an algorithm that guides LLMs to decompose “question-answer” into “question-
855 thinking-answer” through “think step by step” prompting. It has been widely adopted in
856 the field of LLM research.
- **self-consistency CoT:** It is an algorithm that extends CoT. This method enhances the ro-
857 bustness and accuracy of CoT by generating multiple distinct inference paths and selecting
858 the most consistent answer from these paths through a majority voting mechanism.
- **ToT:** This is an algorithm that integrates tree search concepts into the inference process
859 of large language models. It transforms the traditional single-path expansion into a tree-
860 structured exploration framework, with the key idea being that the language model can
861 evaluate multiple intermediate steps during inference through methods such as voting or
862 scoring. These intermediate steps correspond to nodes in the tree, and the inference path is
863

864
865
866
867 Table 7: Sample examples from different datasets belonging to diverse task domains.
868
869
870
871
872
873
874
875
876
877
878

Dataset	Instructions and GT Answers for Each Dataset used to evaluate CoEvo
MMLU	Question: Which statement best explains the purpose of Hart’s distinction between ‘being obliged’ and ‘having an obligation’? Answer: It illuminates the concept of a rule.
CQA	Question: A John is a bum. Much like the stereotype, he lives near this sort of transportation infrastructure. Where does he live? Answer: bridge.
GSM8K	Question: The ratio representing the age of Halima, Beckham, and Michelle is 4:3:7, respectively. If the total age for the three siblings is 126, calculate the age difference between Halima and Beckham. Answer: 9.
WinoGrande	Question: The ring that I bought my girlfriend was worse than the bracelet that I bought her because the ___ was more expensive. Answer: bracelet.

879
880 Table 8: Experiment Setup. Dataset configurations and parameter settings.
881

Attributes	cloud-to-edge stage				edge-to-cloud stage		
	MMLU	CQA	GSM8K	WinoGrande	CQA	GSM8K	WinoGrande
Task size	6MB	3.7MB	4.9MB	8.2MB	3.7MB	4.9MB	8.2MB
Instruction number	13985	10962	8792	43432	10962	8792	43432
Task Scenario	Multi-task	Commonsense	Math	Natural Language	Commonsense	Math	Natural Language
Batch Size	$s = 64$	$s = 32$	$s = 32$	32	$s = 32$	$s = 32$	$s = 32$
Learning Rate	$l = 1e-5$	$l = 1e-5$	$l = 1e-5$	$l = 2e-5$	$l = 1e-5$	$l = 1e-5$	$l = 2e-5$
Local training epoch	$E = 5$	$E = 5$	$E = 3$	$E = 3$	$E = 5$	$E = 3$	$E = 3$

890
891 dynamically optimized through backtracking or heuristic search methods (such as breadth-
892 first search or depth-first search).
893894 **Fine-tuning-based optimization methods:**
895

- **DPO:** It is an algorithm that fine-tunes language models by directly optimizing human preference data. It bypasses the complex reward model training steps of traditional RLHF by reformulating the preference learning problem as a policy-based loss function, which directly maximizes the logarithmic probability difference between preferred and non-preferred responses through a simple binary cross-entropy objective.
- **Distill step by step:** This algorithm first trains a small model to generate intermediate inference steps (rationales), then jointly optimizes them with the final answer prediction. This approach enables the model to learn and acquire the logical inference capabilities of larger models while maintaining its lightweight nature.
- **SPIN:** This algorithm builds upon DPO and enables continual optimization of large models through a self-play mechanism, without relying on additional human preference data or reinforcement learning frameworks. The key lies in modeling the fine-tuning process as a “two-player game” task: the main model (current version) and a historical model (previous version) generate responses to questions, and a contrastive learning objective (similar to DPO) drives the main model to progressively surpass the inference performance of the historical model.

912
913 **C CONFIGURATIONS**
914915 We document the specific settings and implementation details of all experiments mentioned in the
916 main text to ensure the reproducibility of the research. When replicating various baseline methods,
917 we strictly adhere to the design principles outlined in their original papers and implement them
based on publicly available code repositories.

918

Table 9: Performance of CoEvo and baseline methods on other heterogeneous model pairs.

919

920

921

922

923

924

925

926

927

928

929

930

931

932

For efficient inference methods based on CoT: (1) CoT: Standard chain-of-thought inference is triggered using a few-shot prompting. (2) Self-Consistency CoT: For each question, balancing inference overhead and sampling diversity, the model generates five inference paths and selects the most consistent answer through a "majority voting" mechanism. (3) ToT: At each inference step, the model generates three candidate inference paths. Depending on the specific task, the depth of the inference tree (i.e., the maximum number of inference steps) is set between two and three. At each step, three evaluations are performed to filter the optimal inference path.

933

934

935

936

937

938

939

940

941

942

For fine-tuning-based optimization methods: (1) DPO: We use the responses generated by cloud-side LLM as preferred data and randomly select one from the remaining answer options as non-preferred data. (2) Distill Step by Step: The original settings are followed. (3) SPIN: We use the responses generated by cloud-side LLM as the initial SFT data.

943

More experimental details can be found in Table 8 in the Appendix.

944

945

D ADDITIONAL RESULTS

946

947

D.1 EXPERIMENTAL VALIDATION ON OTHER HETEROGENEOUS MODEL PAIRS

948

949

The effectiveness of the CoEvo method is validated on heterogeneous model pairs (Qwen1.7B/14B), with specific results shown in Table 9. Our new experimental results consistently demonstrate that our framework remains effective on the datasets evaluated using the Qwen model pair. This positive outcome strongly supports the universality of our method across different model architectures and scale ratios.

950

951

D.2 COMPARATIVE ANALYSIS OF TEXT PERPLEXITY

952

953

954

Perplexity (PPL) is a fundamental metric for evaluating the performance of language models by measuring their uncertainty in predicting text sequences. Given a sequence of tokens $W = (w_1, w_2, \dots, w_N)$, perplexity is computed as:

955

956

957

958

959

$$PPL(W) = \exp \left(-\frac{1}{N} \sum_{i=1}^N \log P(w_i | w_1, \dots, w_{i-1}) \right). \quad (8)$$

960

where:

961

962

963

964

965

966

967

968

969

970

971

- N : Total number of tokens in the evaluation sequence
- w_i : The i -th token in the sequence
- $P(w_i | w_1, \dots, w_{i-1})$: Conditional probability of token w_i given preceding context
- \log : Natural logarithm (base e)

Lower perplexity values indicate better model performance, with a theoretical minimum of 1 (perfect prediction) and a baseline value equal to the vocabulary size for random guessing. This metric

Table 10: Text perplexity comparison of proposed and baseline methods.

Methods	cloud-to-edge stage				edge-to-cloud stage		
	MMLU	CQA	GSM8K	WinoGrande	CQA	GSM8K	WinoGrande
Meta Llama3 8B	4.49	10.26	6.77	3.47	—	—	—
Meta Llama3 70B	—	—	—	—	7.91	8.43	2.27
Efficient inference based on CoT							
Chain of Thought	4.08	9.44	6.56	3.29	—	—	—
Self-Consistency CoT	4.08	9.44	6.56	3.29	—	—	—
Tree of Thought	3.40	8.08	5.94	3.01	—	—	—
Fine-tuning-based optimization							
DPO	3.37	7.43	5.67	2.87	—	—	—
Distill step by step	3.25	6.82	5.98	3.06	—	—	—
SPIN	3.28	7.52	5.91	3.03	—	—	—
CoEvo	2.90	6.51	5.60	2.59	7.65	7.99	2.22

reflects how "surprised" the model is when encountering the test data, with well-calibrated models achieving lower perplexity on in-distribution text.

Following the experimental setup outlined in Sections 4.1 and 4.2, we evaluate the perplexity of responses generated by CoEvo and baseline approaches across multiple datasets. The test results are summarized in Table 10 in the Appendix.

During cloud-to-edge and edge-to-cloud phase, the perplexity achieved by CoEvo outperforms existing baseline methods on most datasets. Although superior perplexity does not directly equate to correct inference, it reflects the model's clarity in understanding problems and has become one of the widely adopted evaluation metrics in the field of LLMs.

D.3 THE IMPACT OF DIFFERENT OPTIMIZATION OBJECTIVES ON FINE-TUNING OUTCOMES

As mentioned in Section 3.3, we design a joint optimization objective based on three key metrics: answer, rationale, and confidence. Then facilitate the distillation of knowledge from the LLM on the cloud to the SLM on the edge. In this section, we elaborate on the exploration of specific formulations for this joint optimization objective during the experimental phase. We designed three variants of the objective function:

$$\mathcal{L} = \text{conf} \cdot (1 - \alpha) \mathcal{L}_R + \alpha \mathcal{L}_A. \quad (9)$$

where α is a hyperparameter. The design intuition is to use the original confidence score to adjust the final answer. When the confidence is low, the weight of the final answer in the training objective is reduced, guiding the model to focus more on learning the inference process.

$$\mathcal{L} = (1 - \text{conf}) \mathcal{L}_R + \text{conf} \cdot \mathcal{L}_A. \quad (10)$$

The idea here is to dynamically balance the contributions of \mathcal{L}_R and \mathcal{L}_A using the confidence score. When confidence is high, the training objective shifts emphasis toward the rationale.

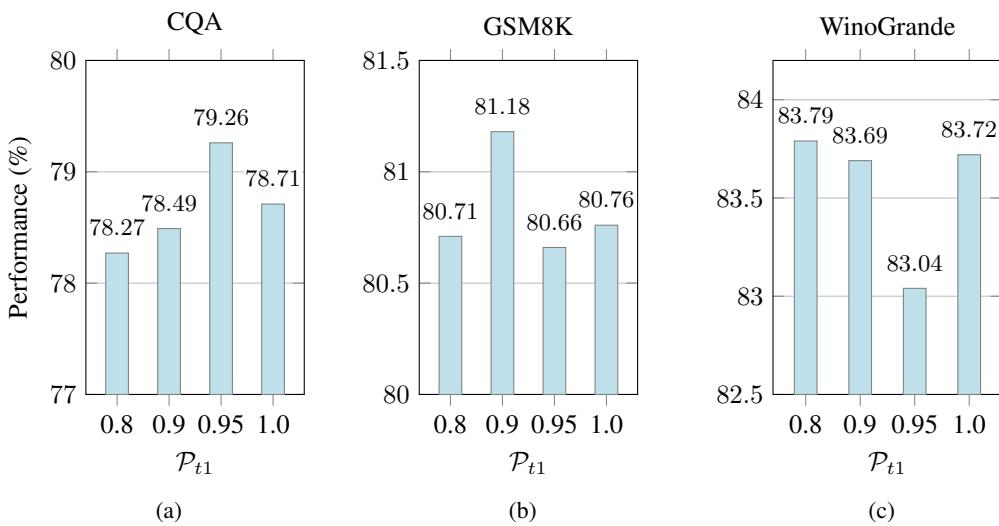
$$\mathcal{L} = \exp(\text{conf}) \cdot [(1 - \alpha) \mathcal{L}_R + \alpha \mathcal{L}_A]. \quad (11)$$

where α is a hyperparameter. This formulation uses the exponentially scaled confidence score to dynamically adjust the weighted sum of \mathcal{L}_R and \mathcal{L}_A .

We evaluate the performance of the optimized SLM with these three variants in various datasets. We denote the SLM obtained by applying Eq. 9 as \mathcal{M}_{1-A} , the SLM obtained by applying Eq. 10 as \mathcal{M}_{1-B} , and the SLM obtained by applying Eq. 11 as \mathcal{M}_{1-C} . We selected distill step by

Table 11: Performance of SLM Variants Optimized with Different Objective Functions.

Methods	MMLU	CQA	GSM8K	WinoGrande
Meta Llama3 8B	57.26	59.38	52.15	62.46
Distill step by step	64.38	65.44	59.31	68.82
\mathcal{M}_{1-A}	65.91	66.26	60.77	68.71
\mathcal{M}_{1-B}	64.28	64.79	59.20	67.88
\mathcal{M}_{1-C}	63.60	63.25	57.11	70.40

Figure 5: Performance comparison under different \mathcal{P}_{t1} thresholds on three datasets. Given the presentation requirements, $\mathcal{P}_{t1}=1$ here actually represents using the complete dataset.

step—which also employs a composite optimization objective utilizing both labels and rationales, but does not incorporate confidence—as the baseline for comparison. The results are shown in Table 11.

\mathcal{M}_{1-A} achieved the best performance on the three datasets: MMLU, CQA and GSM8K, while \mathcal{M}_{1-C} performed optimally on WinoGrande. This indicates that it is meaningful to analyze and construct different forms of optimization objectives tailored to tasks across various domains. It should also be noted that the construction method \mathcal{M}_{1-B} doesn’t achieve satisfactory results on any of the test datasets, indicating that improper utilization of the confidence metric may disrupt its intrinsic relationship with labels and rationales, thereby leading to suboptimal outcomes.

D.4 THE EFFECT OF DATA FILTERING THRESHOLDS ON CLOUD FEEDBACK PERFORMANCE

In this section, we supplement the exploration of a key sub-technique in edge-to-cloud stage of CoEvo: response filtering. The objective of response filtering is to obtain high quality knowledge from \mathcal{M}_2 - the ‘domain-specific expert’ enhanced in cloud-to-edge stage and enhancement of domain-specific data. Specifically, for responses generated by \mathcal{M}_2 based on historical interactions, CoEvo filters qualified high-quality answers according to the design described in Section 3.3, and uploads these to the cloud as raw knowledge for feedback optimization of the cloud-side LLM. We explored the effect of \mathcal{P}_{t1} —that is, the ratio of the maximum probability to \mathcal{P}_{t1} —under different thresholds on the optimization of the cloud-side LLM. Experimental results are shown in Figure 5 in the Appendix.

1080 We sequentially set \mathcal{P}_{t1} to 0.8, 0.9, 0.95, and 1.0. It should be noted that, for ease of presentation,
 1081 $\mathcal{P}_{t1}=1.0$ actually represents using the complete dataset without any filtering. The setting of \mathcal{P}_{t1} ex-
 1082 hibited varying impacts on the optimization of the cloud-side LLM across different domain datasets.
 1083 For CQA, the optimal \mathcal{P}_{t1} was 0.95, while for GSM8K, this value changed to 0.9. For WinoGrande,
 1084 the optimal value was 0.8. We attribute this to the LLM’s differing levels of confidence when ad-
 1085 dressing problems from various domains, which aligns with the observations we obtained in Figure
 1086 2.
 1087
 1088
 1089
 1090

1091 D.5 DESIGN OF TASK-SPECIFIC PROMPT TEMPLATES

 1092
 1093

1094 As discussed in Section 3.2, in cloud-to-edge stage, the primary objective of the cloud-side LLM
 1095 is to provide high-quality composite knowledge to the base SLM \mathcal{M}_0 , specifically including labels,
 1096 rationales, and confidence. The templates used for prompting the generation of each component are
 1097 as follows:

1098 1. Generating Label ($\mathcal{Q} \rightarrow \mathcal{A}$):

1100 "Please complete the specified task according to the requirements in the following task flow.

1101 Task Flow: Read the question below, each of which contains multiple options. Analyze the question
 1102 and directly provide the correct answer. Do not include any content beyond the answer.

1103 Question: [Question]. Your response: {response}"

1104 2. Generating Rationale ($\mathcal{Q} \rightarrow \mathcal{R}$):

1105 "Please complete the specified task according to the requirements in the following task flow.

1106 Task Flow: Read the question below, each of which contains multiple options. Analyze the question
 1107 and provide the inference/thought process for solving this problem. You need to think step by step
 1108 to arrive at the final answer. Do not include any content beyond the inference/thought process.

1109 Question: [Question]. Your response: {response}"

1110 3. Generating Confidence ($\mathcal{Q} \rightarrow \mathcal{C}$):

1111 "Please complete the specified task according to the requirements in the following task flow.

1112 Task Flow: Read the question below, each of which contains multiple options. Analyze the question
 1113 and provide the correct answer along with your confidence level in this answer. The confidence level
 1114 ranges from 1 to 10, representing low to high confidence. Do not include any content beyond the
 1115 correct answer and the confidence level.

1116 Question: [Question]. Your response: {response}"

1117 The [Question] in the prompt templates above refers to the original instruction, the {response}
 1118 represents the response content generated by \mathcal{M}_0 .

1119 As discussed in Section 3.3, in edge to cloud stage, the edge-side SLM \mathcal{M}_2 generates preliminary
 1120 response content based on historical interaction records. Subsequently, CoEvo filters these contents
 1121 to obtain a confidence probability matrix for the feedback to the cloud. The prompt templates used
 1122 for the generation are as follows:

1123 Generating response:

1124 "Please complete the specified task according to the requirements in the following task flow.

1125 Task Flow: Read the question below, each of which contains multiple options. Analyze the question,
 1126 provide the inference/thought process for solving this problem and finally provide the correct answer.
 1127 Please do not provide any content unrelated to the question.

1128 Question: [Question]. Your response: {response}"

Table 12: Qualitative Comparison of Responses Between CoEvo and Best Baseline

Dataset	Instructions and GT Answers for Each Dataset used to evaluate CoEvo
MMLU	<p>Question: Statement 1— The F1 score can be especially useful for datasets with class high imbalance. Statement 2— The area under the ROC curve is one of the main metrics used to assess anomaly detectors.</p> <p>Base: The F1 score can be especially useful for datasets with class high imbalance because it takes into account both precision and recall. The area under the ROC curve is one of the main metrics used to assess anomaly detectors, but it does not take into account class imbalance. Therefore, Statement 1 is true and Statement 2 is false. Answer: True, False. \times</p> <p>Best Baseline: The F1 score is a measure of accuracy that takes into account both precision and recall. It is often used for datasets with class high imbalance, where there are many more examples of one class than another. The area under the ROC curve is a measure of the performance of an anomaly detector. It is not specifically designed for datasets with class high imbalance. Therefore, statement 1 is true, while statement 2 is false. \times</p> <p>CoEvo: Both statements are true. The F1 score is often used to evaluate the performance of binary classifiers, while the area under the ROC curve is commonly used to assess the performance of anomaly detectors. Both metrics can be particularly helpful when dealing with imbalanced datasets, as they provide a more comprehensive view of the model's performance than accuracy alone. \checkmark</p>
CQA	<p>Question: A John is a bum. Much like the stereotype, he lives near this sort of transportation infrastructure. Where does he live?</p> <p>Base: The question tests the ability to recognize a stereotype (John as a bum). - Key clues: "stereotype", "transportation infrastructure"(industrial context) - Best fit: (C)train station (connotes a low-income area) - Why not others: (A)/(B)/(D)/(E) are plausible but don't match the bum stereotype. \times</p> <p>Best Baseline: The key is to recognize a transportation hub. - Key clues: "bum", "transportation infrastructure" (context of travel) - Best fit: (C)train station (central hub) - Why not others: (A)bus depot is plausible but secondary (focus is on trains) (B)beach is contextually irrelevant (D)/(E) are incorrect due to incorrect spelling. \times</p> <p>CoEvo: The context implies a homeless person, suggesting proximity to urban services. - Key clues: "bum", "near this sort of transportation infrastructure" (impoverished, city-based) - Best fit: (D)bridge (a common location for homeless encampments) - Why not others: (B)beach is unlikely due to remoteness (A)/(C) are plausible but less specific (E)bridge is a red herring (a bridge is a bridge). \checkmark</p>
GSM8K	<p>Question: The ratio representing the age of Halima, Beckham, and Michelle is 4:3:7, respectively. If the total age for the three siblings is 126, calculate the age difference between Halima and Beckham.</p> <p>Base: $4x+3x+7x=126$. $14x=126$. $x=126/14$. $x=9$. Age difference between Halima and Beckham=7-3. Age difference between Halima and Beckham=4. \times</p> <p>Best Baseline: Let x be the age of Halima. Then, the age of Beckham is $3x$, and the age of Michelle is $7x$. The sum of the three ages is 126. So, $x + 3x + 7x = 126$ $11x = 126$ $x = 126 / 11 = 11.5$ The age of Halima is 11.5 years old. The age of Beckham is $3x = 3 * 11.5 = 34.5$ years old. Therefore, the age difference between Halima and Beckham is $34.5 - 11.5 = 23$ years. \times</p> <p>CoEvo: The sum of the three numbers in the ratio is $4 + 3 + 7 = 14$. Halima's age is $4/14$ of the total age, or $(4/14) * 126 = 36$ years old. Beckham's age is $3/14$ of the total age, or $(3/14) * 126 = 27$ years old. The age difference between Halima and Beckham is $36 - 27 = 9$ years. \checkmark</p>
WinoGrande	<p>Question: William liked wearing high heels when going out with Lawrence, because ... loved the feeling of heels.</p> <p>Base: The sentence states that William liked wearing heels when going out with Lawrence. It also states that Lawrence loved the feeling of heels. This implies that William wears heels for Lawrence's sake. Therefore, the blank should refer to Lawrence. \times</p> <p>Best Baseline: William liked wearing high heels with Lawrence. William's reason for wearing heels was that Lawrence loved the feeling of heels. Therefore, the blank should refer to Lawrence. \times</p> <p>CoEvo: The sentence states that William likes wearing high heels when going out with Lawrence, because someone loves the feeling of heels. Since liking the feeling of heels would lead to wearing high heels, the blank should refer to William. \checkmark</p>

D.6 QUALITATIVE DIFFERENCES IN GENERATED INFERENCE PATHS

Table 12 in the Appendix presents a comparison of the response quality generated by CoEvo and the best baseline method across different domain datasets in the cloud-to-edge stage. By leveraging its credible CoT construction approach, CoEvo achieves higher-quality responses than best baseline .