

# BCL: Bayesian In-Context Learning Framework for Information Extraction

Haoliang Liu<sup>1\*</sup> Chengkun Cai<sup>2\*</sup> Xu Zhao<sup>3\*</sup> Han Zhu<sup>4</sup> Shizhou Huang<sup>5†</sup>  
Xinglin Zhang<sup>6</sup> Tao Chen<sup>7</sup> Jenq-Neng Hwang<sup>8</sup> Zhang Huaping<sup>9</sup> Lei Li<sup>9‡</sup>

## Abstract

Existing information extraction (IE) tasks increasingly adopt in-context learning (ICL) with large language models. However, current approaches either show inconsistent performance across model scales or lack systematic optimization and generalizability. Building on this, we propose BCL (Bayesian In-Context Learning Framework for Information Extraction), the first optimization framework that uses particle filtering with Bayesian updates to systematically refine label representations across IE tasks. Through four steps—initialization, observation, weight update, and resampling, BCL generalizes to both sequence labeling and relation classification paradigms. Extensive experiments demonstrate substantial and consistent improvements over existing approaches.

## 1 Introduction

Recent IE tasks rely on in-context learning (ICL), where large language models (LLMs) (Brown et al., 2020) are guided by contextual information. Recent approaches can be broadly categorized into task transfer approaches that reformulate information extraction (IE) as auxiliary tasks (e.g., ChatIE (Wei et al., 2023), CodeIE (Li et al., 2023a)) and guideline-based approaches that provide explicit annotation guidelines (e.g., GuideNER (Huang et al., 2025)).

However, existing approaches face practical limitations. As illustrated in Figure 1 (top), task transfer methods show inconsistent performance across model scales. While potentially effective on ultra-large commercial models, they often underperform direct IE prompting on smaller models. ChatIE

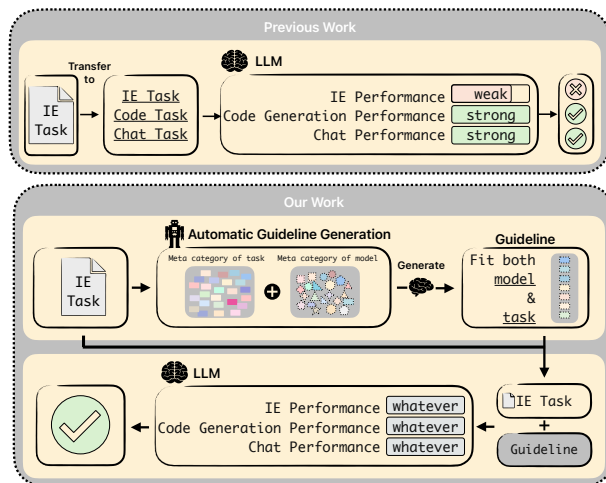


Figure 1: **Comparison of previous approaches and our work.** **Top:** Previous methods convert IE tasks to leverage models’ stronger code or chat capabilities, making performance dependent on model-specific strengths. **Bottom:** Our approach directly improves IE performance via automatically generated semantic patterns, regardless of models’ relative strengths across task types.

underperforms one-shot prompting by a substantial margin on NER tasks, and CodeIE fails on RE tasks with near-zero micro-F1. This inconsistency makes deployment challenging when using lightweight models, which are common in practical settings due to computational constraints.

Guideline-based approaches offer an alternative to task transfer methods, but existing work has critical limitations. GuideNER (Huang et al., 2025), the current state-of-the-art, has significant limitations. First, it uses simple frequency-based selection without systematic optimization for guideline quality. Second, it is designed specifically for NER and does not extend to other IE tasks, as evidenced by the absence of RE results in Figure 2. These limitations motivate the need for a more general and optimized approach.

Building on these observations, we introduce automatic subcategory generation (Figure 1, bottom)

\* Equal contribution. † Important contribution.

‡ Corresponding authors: lilei@bit.edu.cn

<sup>1</sup>HiThink Research <sup>2</sup>University College London

<sup>3</sup>University of Edinburgh

<sup>4</sup>The Hong Kong University of Science and Technology

<sup>5</sup>East China Normal University

<sup>6</sup>Shanghai Medical Image Insights

<sup>7</sup>University of Waterloo <sup>8</sup>University of Washington

<sup>9</sup>Beijing Institute of Technology

**Performance Comparison Across Methods and Task Types (Qwen-2.5-7B Model)**

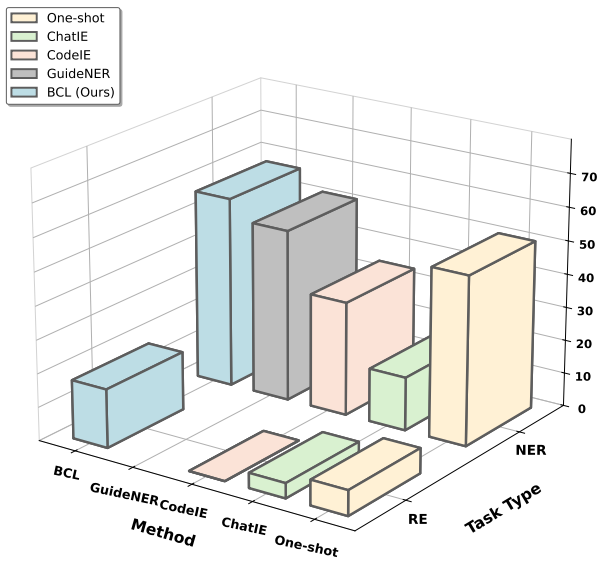


Figure 2: Performance comparison of different methods on Qwen-2.5-7B across NER and RE tasks. The Y-axis represents F1 score (%). BCL demonstrates consistent superiority over baseline methods, while ChatIE and CodeIE show substantial degradation on both task types. GuideNER is applicable only to NER tasks.

that decomposes labels into semantically discrete atomic representations. The key insight is that IE labels are often coarse-grained: a "Person" label in NER could mean family roles like "father" or "friend" to the model’s prior understanding, while in a specific dataset it may only refer to public figures such as "athlete" or "politician". To bridge this gap between the model’s prior knowledge and the dataset’s annotation schema, we represent each label using multiple subcategories as atomic representations, with these subcategory patterns serving as rules to clarify the label’s specific meaning in context. For example, in NER, "Person" can be represented by subcategories such as "athlete" and "public figure"; in RE, "[FRESNO, Located-In, Calif]" can be decomposed into "[city, spatial-containment, state]" and "[subregion, asymmetry, super-region]". Crucially, by discretizing labels into semantic atomic units, we can treat them as controllable discrete variables. This enables us to optimize label representations using optimization algorithms such as particle filtering, where each rule is a particle with an associated weight, refined through iterative evaluation and Bayesian updates.

We introduce **BCL (Bayesian In-Context**

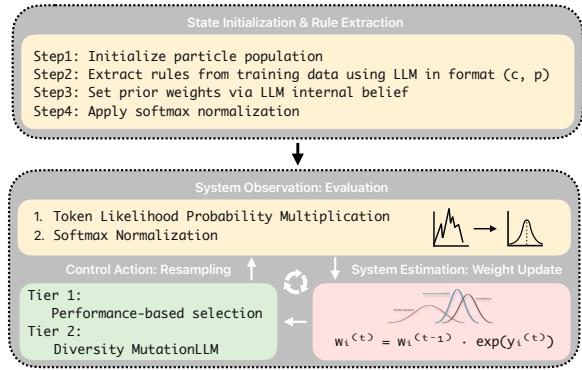


Figure 3: Overview of the BCL framework. The framework operates through particle filtering with Bayesian updates, alternating between observation and control to progressively optimize the semantic patterns distribution.

**Learning Framework for Information Extraction**), optimizing subcategory patterns through four steps (Figure 3): (1) initialization—generate initial patterns and set prior weights, (2) observation—evaluate via ICL-based IE to compute likelihoods, (3) weight update—refine weights through Bayesian update, (4) resampling—eliminate low-weight particles and diversify high-performers via LLM mutation.

Our contributions are:

- We introduce the key insight of treating context as controllable discrete variables, achieved by decomposing labels into fine-grained semantic units, enabling systematic optimization methods to be applied.
- We develop the first optimization framework using particle filtering with Bayesian updates that generalizes across IE tasks, achieving systematic quality improvement on both sequence labeling and relation classification paradigms.
- Extensive experiments demonstrate substantial improvements over existing approaches (up to 30%), achieving strong performance while other methods either fail to generalize or show limited effectiveness.

## 2 Related Work

### 2.1 In-Context Learning for Information Extraction

In-context learning (ICL) enables large language models to adapt to new tasks through demonstration examples without parameter updates (Brown

et al., 2020; Wei et al., 2022; Min et al., 2022). Traditional ICL approaches for information extraction rely on example-based demonstrations, where models learn input-output mappings through pattern recognition (Dong et al., 2022; Li et al., 2023b).

Recent work improves ICL for IE by refining demonstration construction, retrieval, and filtering strategies. C-ICL (Mo et al., 2024) incorporates both positive and hard negative examples into demonstrations, while G&O (Li et al., 2024b) decomposes generation into intermediate reasoning and structured outputs to improve stability. GuideNER (Huang et al., 2025) replaces demonstrations with LLM-generated annotation guidelines, and Dr.ICL (Luo et al., 2024) retrieves task-relevant examples to enhance reasoning performance. Similarly, MAPS (Chen et al., 2025) introduces anchor-based sampling for fine-grained entity linking, while recent LLM-based feature selection methods (Wang et al., 2025) further highlight the importance of iterative filtering for structured extraction. Related observations also appear in adjacent multimodal understanding settings: Human Motion Instruction Tuning (Li et al., 2025) and Multiple Human Motion Understanding (Li et al., 2026) show that carefully designed instruction and structured semantic supervision can improve complex motion understanding, suggesting that input organization and guidance are broadly important for structured prediction.

Beyond demonstration design, recent studies analyze the intrinsic mechanisms of ICL and context utilization. Shi et al. (2026) study entropy in context length scaling, while Cai et al. (2025b) examine the roles of deductive and inductive reasoning. Lan et al. (2025) further propose attention consistency to estimate token importance, providing insights into how models utilize demonstrations during inference.

For relation extraction, ICL faces challenges in modeling inter-entity dependencies and contextual patterns. GPT-RE (Wan et al., 2023) retrieves task-aware demonstrations with label-guided reasoning, while Li et al. (2024a) propose a recall-retrieve-reason framework to enhance retrieval and reasoning. Wadhwa et al. (2023) highlight performance variance across prompts, and CodeIE (Li et al., 2023a) reformulates IE as code generation but remains sensitive to demonstration quality.

Recent studies show that LLMs can perform structured reasoning in complex settings.

CountLLM (Yao et al., 2025) highlights structured dependency modeling, while other work explores retrieval-reasoning in multi-hop QA (Ji et al., 2026), few-shot generalization without explicit meta-learning (Guan et al., 2025; Guan, 2025), and structured context in visually grounded retrieval-augmented generation (Ji et al., 2025). Together, these findings highlight the importance of context utilization.

## 2.2 Control-Theoretic and Probabilistic Optimization

Classical control theory (Åström and Murray, 2021) models complex systems as input-output mappings governed by feedback mechanisms, where external control variables can systematically steer system behavior without directly observing internal states. Particle filtering (Gordon et al., 1993) and sequential Monte Carlo methods (Doucet et al., 2001) estimate latent states in high-dimensional nonlinear systems via population-based sampling and importance resampling. In black-box optimization, Bayesian optimization (Frazier, 2018; Xu et al., 2026) builds probabilistic surrogate models with acquisition functions to guide sampling, while Approximate Bayesian Computation (Beaumont et al., 2002; Liu, 2026) enables likelihood-free inference for complex models. These approaches share a common principle: optimizing system behavior via input-output observations without access to internal mechanisms. Evolutionary prompt optimization (Qi et al., 2024) applies population-based search to LLM behavior, but lacks systematic control-theoretic grounding and focuses on reasoning tasks rather than structured prediction. In contrast, our work integrates control-theoretic principles with sequential Monte Carlo methods to optimize demonstration selection in few-shot learning.

Recent work begins to apply such principles to controlling LLM behavior. For example, Cai et al. (2025a) leverage Bayesian optimization to steer LLM-driven image editing processes under black-box settings, demonstrating the effectiveness of probabilistic search for controllable generation.

## 2.3 Optimization Approaches for LLM Behavior

Various optimization strategies have been explored for LLM behavior control (Zhao et al., 2026; Cao and Zhao, 2025). Fine-tuning (Wei et al., 2021) requires substantial resources and labeled data, lim-

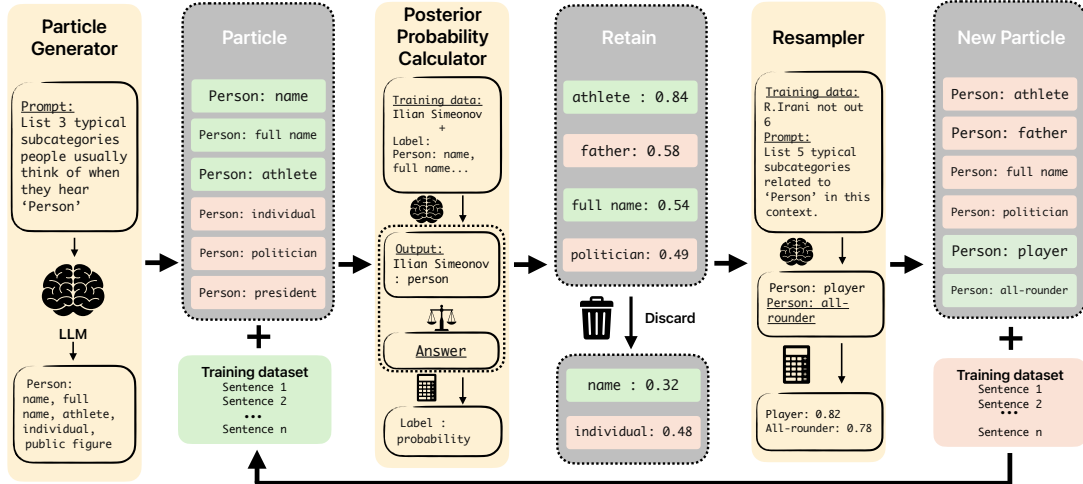


Figure 4: Overall framework of BCL showing the particle-based rule optimization pipeline with iterative generation (Particle Generator), evaluation (Posterior Probability Calculator), selection (Retain), and mutation (Resampler) phases guided by LLM performance feedback.

iting few-shot applicability. Prompt engineering (Zhou et al., 2022; Pryzant et al., 2023) relies on manual effort or local search heuristics, while evolutionary algorithms (Qi et al., 2024) explore prompt spaces but lack systematic guideline optimization for structured prediction. Existing methods rely on heuristic strategies without control-theoretic grounding (Zhao et al., 2021). Although RLHF (Ouyang et al., 2022) and preference optimization (Rafailov et al., 2023) address alignment, they modify model parameters rather than optimizing external control inputs like demonstration selection rules.

### 3 Methodology: BCL

Our BCL framework consists of a comprehensive control-theoretic approach for rule optimization, as illustrated in Figure 3 and Figure 4. Figure 3 shows the algorithmic overview with the four key steps of our adaptive filtering process, while Figure 4 presents the particle-based optimization pipeline with iterative generation, evaluation, selection, and mutation phases.

#### 3.1 Problem Formulation

Given a pre-trained large language model  $\mathcal{M}$  and a target dataset  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$  with training split  $\mathcal{D}_{train}$ , development split  $\mathcal{D}_{dev}$  and test split  $\mathcal{D}_{test}$ , our goal is to find an optimal rule list  $\mathcal{R}^*$  that maximizes the model’s information extraction performance on the development set, and then evaluate its generalization performance on the test set.

#### 3.1.1 Dataset-Specific Optimization

Since different datasets have varying data distributions and annotation conventions, we need to adapt the rule selection to each specific dataset. Using the training portion  $\mathcal{D}_{train}$ , we extract candidate rules, and then optimize the rule list  $\mathcal{R}^*$  based on performance on the development set  $\mathcal{D}_{dev}$ . This ensures that the final rule list is tailored to both the dataset characteristics and the specific LLM’s behavior.

#### 3.1.2 ICL-based Extraction

For any input  $x$ , the extraction process follows:

$$\hat{y} = \mathcal{M}(\text{Prompt}(x, R_t)) \quad (1)$$

where  $\mathcal{M}$  is the pre-trained LLM,  $\text{Prompt}(x, R_t)$  constructs the input prompt by combining text  $x$  with the optimal rule configuration  $R_t$ , and  $\hat{y}$  is the predicted extraction output.

#### 3.1.3 Optimization Objective

We seek to find the optimal rule configuration that maximizes performance on unseen data. Following standard machine learning practice, we split the available training data into training and validation sets, and optimize rules based on validation performance:

$$R_t^* = \arg \max_{R_t} \frac{1}{|\mathcal{D}_{val}|} \sum_{(x,y) \in \mathcal{D}_{val}} F(\hat{y}, y) \quad (2)$$

where  $R_t = \{(p_i^{(t)}, c_i, w_i^{(t)})\}_{i=1}^N$  represents a rule configuration with  $N$  particles at iteration  $t$ , where  $p_i^{(t)}$  is the  $i$ -th subcategory pattern (particle),  $c_i$  is its corresponding entity label, and  $w_i^{(t)}$  is the confidence weight,  $F(\cdot, \cdot)$  is a performance metric (e.g., F1 score). The rule extraction and initial population are derived from  $\mathcal{D}_{\text{train}}$ , while the optimization objective is evaluated on the held-out validation set  $\mathcal{D}_{\text{val}}$  to prevent overfitting. Final evaluation is performed on  $\mathcal{D}_{\text{test}}$ .

### 3.1.4 Challenges with Existing Approaches

While existing rule-based ICL methods like GuideNER have demonstrated the effectiveness of annotation guidelines over examples, they lack systematic optimization strategies for rule selection and combination. Current approaches rely on heuristic frequency-based filtering, which fails to capture the complex interdependencies between rules and their collective impact on IE performance.

### 3.1.5 Control-Theoretic Reformulation

Traditional LLM optimization faces a fundamental challenge: the massive parameter space (billions of parameters) renders direct observation and control intractable. We address this by treating rules as a low-dimensional, observable interface to the LLM system.

Recent findings suggest that in-context learning operates as a rule-based inference system, where the quality of rules—not examples—determines performance. This motivates reformulating the optimization as a control system problem, where rules serve as externally controllable state variables:

$$R_{t+1} = f(R_t, u_t) \quad (\text{Rule Evolution}) \quad (3)$$

$$y_t = h(R_t) \quad (\text{Performance Observation}) \quad (4)$$

where  $R_t$  represents the rule configuration at iteration  $t$ ,  $u_t$  denotes control actions (rule modifications), and  $y_t$  is the observed IE performance.

## 3.2 Adaptive Rule Filtering Algorithm

The discrete, combinatorial nature of rule spaces and nonlinear performance mappings makes classical control methods inappropriate. We develop an adaptive filtering approach that iteratively estimates optimal rule configurations through performance feedback.

As illustrated in Figure 3, our approach follows a systematic particle filtering process:

### 3.2.1 Particle-based State Representation

**Terminology.** To clarify key concepts used throughout this section:

- **Particle:** A subcategory pattern paired with its entity label, e.g., (“athlete”, Person)
- **Rule:** A particle with its associated confidence weight in the population
- **Weight:** Normalized probability  $w_i \in [0, 1]$  indicating particle quality, where  $\sum_i w_i = 1$

At time step  $t$ , the complete rule configuration is:

$$R_t = \{(p_i^{(t)}, c_i, w_i^{(t)})\}_{i=1}^N \quad (5)$$

where  $p_i^{(t)}$  is the  $i$ -th subcategory pattern (particle),  $c_i$  is its corresponding entity label, and  $w_i^{(t)}$  is the confidence weight.

The filtering process consists of four steps:

**0. Initialization (Rule Extraction):** We initialize the particle population by extracting initial rules from the training dataset using the GuideNER approach (Huang et al., 2025). For each input-label pair  $(x_j, y_j)$  in the training set, we use the LLM to summarize rule patterns:

$$p_i^{(0)} = \text{LLM}_{\text{extract}}(x_j, y_j, \text{prompt}_{\text{summary}}) \quad (6)$$

The initial particles are assigned prior weights based on their linguistic naturalness under the LLM’s language model. We generate  $N$  particles per entity label (typically  $N = 10$ ) and compute their prior scores:

$$s_i = -\text{PPL}(p_i^{(0)}), \quad i = 1, \dots, N \quad (7)$$

$$w_i^{(0)} = \frac{\exp(s_i)}{\sum_{j=1}^N \exp(s_j)} \quad (8)$$

where perplexity is computed as:

$$\text{PPL}(p_i^{(0)}) = \exp \left( -\frac{1}{|p_i^{(0)}|} \sum_{j=1}^{|p_i^{(0)}|} \log P(t_j | t_{<j}) \right) \quad (9)$$

Lower perplexity indicates the rule text is more natural and coherent according to the model’s internal knowledge, thus receiving higher prior weight.

**1. Evaluation (ICL-based Observation):** Each particle is evaluated through ICL-based IE inference on a validation batch  $\mathcal{B}^{(t)}$  sampled from  $\mathcal{D}_{\text{train}}$ .

For each particle, we compute their confidence scores:

$$s_i^{(t)} = L_c(p_i^{(t)}, \theta), \quad i = 1, \dots, N \quad (10)$$

$$y_i^{(t)} = \frac{\exp(s_i^{(t)})}{\sum_{j=1}^N \exp(s_j^{(t)})} \quad (11)$$

where  $L_c(p_i^{(t)}, \theta)$  computes the average log probability when the LLM uses rule  $p_i^{(t)}$  to generate a label sequence  $T_i$  on a validation sample:

$$L_c(p_i, \theta) = \frac{1}{|T_i|} \sum_{j=1}^{|T_i|} \log P(t_j | t_{<j}, p_i, \theta) \quad (12)$$

Here  $\theta$  represents the pretrained LLM parameters. This provides a length-normalized confidence score  $y_i^{(t)} \in (0, 1]$  reflecting how well the rule performs on the actual IE task.

**2. Weight Update (Bayesian Posterior):** Particle weights are updated by combining prior knowledge with observed performance:

$$\tilde{w}_i^{(t)} = w_i^{(t-1)} \cdot \exp(\beta \cdot y_i^{(t)}) \quad (13)$$

$$w_i^{(t)} = \frac{\tilde{w}_i^{(t)}}{\sum_{j=1}^N \tilde{w}_j^{(t)}} \quad (14)$$

where  $w_i^{(t-1)}$  is the prior weight and  $\exp(\beta \cdot y_i^{(t)})$  rewards higher performance.

**3. Multi-level Resampling with Rule Mutation:** We employ a two-tier strategy balancing exploitation and exploration, adapting the approach from Qi et al. (2024):

**Tier 1 (Performance Selection):** Retain the top 50% of particles by weight, removing low-performing ones.

**Tier 2 (Diversity Mutation):** Apply semantic mutations to retained particles through LLM-guided generation:

$$p_i' = \text{LLM}_{\text{mutate}}(p_i^{(t-1)}, x^{(t)}) \quad (15)$$

where  $x^{(t)}$  provides context. Three strategies are employed: *refinement* (increase specificity), *generalization* (increase coverage), and *contextualization* (generate domain-specific variants).

New particles inherit labels from parents and receive weights based on perplexity as described in Equation 9:

$$s_i' = -\text{PPL}(p_i'), \quad i = 1, \dots, M \quad (16)$$

$$w_i' = \frac{\exp(s_i')}{\sum_{j=1}^M \exp(s_j')} \quad (17)$$

Lower perplexity indicates greater consistency with the model’s language patterns (See Appendix E.1 for the rationale behind our prior and likelihood selection in the Bayesian filtering framework.). The updated configuration combines retained and new particles:  $R^{(t)} = \mathcal{R}_{\text{keep}} \cup \mathcal{R}_{\text{new}}$ .

The iteration continues until: (a) performance saturates with relative improvement  $(F_1^{(t)} - F_1^{(t-3)})/F_1^{(t-3)} < 0.03$  over 3 iterations, or (b) all training samples are exhausted.

As illustrated in Table 4, convergence typically occurs with only 3–5% of training data when performance plateaus.

## 4 Experiment

We conduct extensive experiments on multiple Information Extraction tasks. Following previous work (Huang et al., 2025; Wei et al., 2023; Li et al., 2023a), we use entity-level F1 score for evaluation. While our framework is applicable to various IE paradigms, we focus on two representative tasks: sequence labeling (NER) and relation classification (RE). For NER, this requires correct boundary detection and type classification; for RE, correct identification of both entity arguments and their relation type (see Appendix E.2 for precise definitions). Following GuideNER (Huang et al., 2025), we measure token cost as the average number of input and output tokens per sample, reflecting computational overhead and inference latency.

### 4.1 Datasets

We evaluate on six widely-used IE benchmarks spanning multiple domains and task formulations. For NER, we use CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) (news, 4 entity types), ACE 2005 (Walker et al., 2006) (news/conversational, 7 types), and GENIA (Kim et al., 2003) (biomedical, 5 types). For RE, we use NYT (Riedel et al., 2010) (news, 24 relation types), CoNLL04 (Roth and Yih, 2004) (general domain, 5 types), and SciERC (Zhang et al., 2024) (scientific papers, 7 types). Appendix A provides detailed statistics.

### 4.2 Experiments Setup

All experiments are conducted on a computing cluster equipped with H100 GPUs for computational acceleration. They are implemented using PyTorch 2.6.0 and transformers 4.51.3. We employ four foundation models with varying scales, languages, and training datasets: Qwen2.5-3B, Qwen2.5-7B,

Method	Model	NER			RE			Token Cost
		CoNLL03	ACE05	GENIA	NYT	CoNLL04	SciERC	
One-shot	Qwen-2.5-3b	60.55	27.08	46.78	0.29	19.28	5.01	385
	Qwen-2.5-7b	62.82	35.73	52.91	0.28	28.10	5.54	
	Llama-3.1-8b	65.78	40.98	51.29	0.44	22.57	7.18	
	Pixtral-12B	60.25	39.15	49.73	0.38	21.70	7.86	
ChatIE	Qwen-2.5-3b	39.26	15.41	33.32	0.00	11.11	0.00	942
	Qwen-2.5-7b	25.64	11.83	10.19	0.00	12.48	0.88	
	Llama-3.1-8b	55.12	26.91	35.20	0.00	12.62	1.03	
	Pixtral-12B	60.85	29.28	40.12	0.40	14.50	4.40	
CodeIE	Qwen-2.5-3b	45.92	23.70	6.59	0.00	0.00	0.00	1172
	Qwen-2.5-7b	60.00	22.76	19.17	0.00	0.00	0.00	
	Llama-3.1-8b	0.05	0.06	0.00	0.00	0.00	0.00	
	Pixtral-12B	53.39	18.57	20.62	0.00	0.00	0.00	
GuideNER	Qwen-2.5-3b	63.32	27.43	41.49	—	—	—	506
	Qwen-2.5-7b	65.10	41.57	47.43	—	—	—	
	Llama-3.1-8b	61.38	44.97	42.86	—	—	—	
	Pixtral-12B	64.76	37.64	48.03	—	—	—	
BCL	Qwen-2.5-3b	65.12	35.46	46.98	0.31	35.32	8.28	501
	Qwen-2.5-7b	72.83	46.94	51.36	0.43	42.46	9.57	
	Llama-3.1-8b	69.14	53.10	50.15	0.91	38.93	12.06	
	Pixtral-12B	65.54	42.87	50.65	0.60	25.75	10.34	

Table 1: Performance comparison of different methods (One-shot, ChatIE, CodeIE, GuideNER, and BCL) across Qwen-2.5, Llama-3.1, and Pixtral models on NER (CoNLL03, ACE05, GENIA) and RE (NYT, CoNLL04, SciERC) benchmarks. All results are statistically significant ( $p < 0.05$ ). Gray shading highlights the best-performing method for each model configuration.

Llama3.1-8B and Pixtral-12B. This selection allows us to investigate the impact of different model size on our method’s performance. All models are tested with temperature set to 0.0 and random seed fixed at 42 to ensure reproducibility of the experiments. In addition, during the computation of prior probabilities, all models are evaluated in eval mode to disable dropout and other stochastic behaviors.

For optimization framework, we set the number of particles to 10 based on empirical experience from preliminary experiments (Huang et al., 2025), balancing computational efficiency and exploration capability. The number of data used in each observation step is determined through grid search over the set [1, 3, 5, 7, 9, 11, 13, 15], selecting the value that yields optimal performance for each dataset. This configuration ensures consistent experimental conditions and enables direct performance comparison across different models and datasets.

### 4.3 Baselines

We compare against widely-used ICL approaches in the IE domain.

Our baselines include: **One-shot**, where models perform IE with a single demonstration example to illustrate the task format and desired output struc-

ture. **ChatIE** (Wei et al., 2023) and **CodeIE** (Li et al., 2023a), two task transfer methods that reformulate IE as dialogue or code generation tasks, respectively. Both methods employ sophisticated example selection strategies and are applicable to both NER and RE tasks. **GuideNER** (Huang et al., 2025), the current state-of-the-art guideline-based method for NER, which uses frequency-based rule selection to assist in-context learning. Since GuideNER is specifically designed for NER tasks, we only evaluate it on NER datasets and mark it as "—" for RE tasks.

This experimental design ensures all baseline methods operate under the same paradigm of pure in-context learning, enabling fair comparison of different approaches’ effectiveness.

### 4.4 Main Results

Table 1 shows BCL consistently outperforms all baselines across IE benchmarks and model scales. Task transfer baselines (ChatIE, CodeIE) exhibit severe degradation on smaller models: ChatIE achieves only 25.64 F1 on CoNLL03 (Qwen-2.5-7B) versus BCL’s 72.83, while CodeIE drops to 0.05 F1 (Llama-3.1-8B) versus BCL’s 69.14, reflecting their reliance on model-specific capabilities. Against the rule-based GuideNER, BCL

Table 2: Ablation study on different components of the proposed method.

Variant	F1 Score	Particles
BCL (Full)	72.83	10
Weight Update	61.78	10
Tier-1 Resampling	71.29	10,441
Tier-2 Resampling	62.98	10

shows consistent advantages (72.83 vs. 65.10 on CoNLL03; 51.36 vs. 47.43 on GENIA), demonstrating systematic Bayesian optimization’s superiority over frequency heuristics.

BCL’s advantage amplifies on RE tasks where prior methods fail completely. ChatIE and CodeIE achieve 0.00 F1 across most RE configurations, while GuideNER is inapplicable (marked “—”). In contrast, BCL maintains effective performance (42.46 F1 on CoNLL04; 12.06 F1 on SciERC), demonstrating successful generalization across both NER and RE through adaptive rule optimization.

This stability enables cost-efficient deployment: Qwen-2.5-3B with BCL (65.12 F1) matches Llama-3.1-8B’s one-shot performance (65.78 F1) on CoNLL03, achieving comparable results with 62% fewer parameters. BCL thus represents the first ICL optimization framework leveraging Bayesian inference for adaptive optimization, achieving consistent performance across diverse IE paradigms and model scales while overcoming the task-specific limitations of prior methods. (Appendix B illustrates the evolution process through examples).

#### 4.5 Ablation Study

We conduct ablation studies to validate the contribution of each component in BCL’s particle filtering framework. Table 2 presents results on CoNLL03 with Qwen-2.5-7B, where we systematically remove each mechanism while keeping others intact.

**Bayesian Weight Update.** Removing Bayesian weight updates causes the most significant performance drop (-11.05 F1 points), reducing F1 from 72.83 to 61.78. Without this mechanism, the framework degenerates to random search without principled belief updates, demonstrating that Bayesian inference is the core component enabling BCL’s effectiveness.

**Tier-2 Resampling (Diversity Mutation).** Disabling diversity mutation leads to the second-

largest degradation (-9.85 F1 points), with F1 dropping to 62.98. This confirms that maintaining particle diversity through controlled mutation is crucial for preventing premature convergence to suboptimal solutions and exploring the prompt space effectively.

**Tier-1 Resampling (Performance Filtering).** While removing this component causes minimal performance loss (-1.54 F1 points), particle count explodes from 10 to 10,441, revealing that Tier-1 primarily ensures efficiency by pruning low-quality particles with negligible performance cost.

These results validate BCL’s design: Bayesian weight updates and diversity mutation are essential for performance, while performance filtering ensures efficiency by maintaining a compact particle set without performance degradation.

**Effect of Semantic Decomposition.** To isolate the role of semantically coherent subcategories, we conduct an additional ablation where subcategories are randomly reassigned to different entity labels, breaking semantic alignment while keeping the subcategory pool unchanged. Results (Appendix C) show that this leads to consistent performance degradation, confirming that semantic coherence is critical for effective optimization.

#### 4.6 Cross-Model Generalization

While our method is designed to optimize inference on a given model, it is also important to understand whether the learned rules capture transferable patterns that extend beyond a specific backbone. To this end, we study the cross-model generalization ability of the proposed approach.

We consider a setting where rules optimized on smaller open-source models (e.g., Llama-3.1-8B and Qwen-2.5-7B) are directly applied to a stronger closed-source model, GPT-3.5-turbo. This setup reflects a practical scenario in which optimization is performed on accessible models and then deployed on more capable systems. The inference pipeline remains unchanged, and no additional adaptation is introduced.

As shown in Table 3, rules learned on smaller models consistently yield performance gains when transferred to GPT-3.5-turbo. In particular, the transferred rules outperform the direct GPT-3.5-turbo baseline on both datasets. This suggests that the proposed method captures generalizable reasoning patterns rather than relying on model-specific behaviors.

Overall, these results indicate that the learned

Summary Model	Inference Model	CoNLL03	ACE05
–	GPT-3.5-turbo	73.44	51.03
Llama-3.1-8B	Llama-3.1-8B	<b>69.14</b>	<b>53.10</b>
Llama-3.1-8B	GPT-3.5-turbo	<b>74.60</b>	<b>52.50</b>
Qwen-2.5-7B	Qwen-2.5-7B	<b>72.83</b>	<b>46.94</b>
Qwen-2.5-7B	GPT-3.5-turbo	<b>75.80</b>	<b>53.00</b>

Table 3: Cross-model generalization results. Rules optimized on smaller open-source models can be directly transferred to GPT-3.5-turbo and consistently improve performance.

optimization strategies are not limited to the source model, but can extend to stronger models without additional tuning.

#### 4.7 Parameter Sensitivity Analysis

We perform extensive parameter analysis on two key factors: context window length and training data size.

##### 4.7.1 Impact of Training Data Quantity on BCL Performance

Table 4: F1 scores (%) for different training set sizes.

Training Set Size	CoNLL03	GENIA	ACE05
1%	69.12	44.32	40.32
3%	71.16	50.12	44.04
5%	70.78	51.20	46.94
10%	71.12	51.36	44.43
20%	72.04	49.58	44.89
30%	71.37	50.18	44.98
40%	72.83	49.18	44.62
50%	71.81	49.35	44.27

We investigate BCL’s data efficiency by evaluating performance across varying training set sizes (1% to 50%). Table 3 presents F1 scores on Qwen-2.5-7B across three datasets with varying training set sizes. The results reveal a striking pattern: performance rapidly saturates with minimal data. On CoNLL03, BCL reaches 69.12 F1 with only 1% of training data and peaks at 72.83 F1 with 40%, showing modest improvement (3.7 points) when scaling from 1% to the optimal setting. Similar trends appear on GENIA (peak at 5-10%: 51.20-51.36 F1) and ACE05 (peak at 5%: 46.94 F1). This data efficiency stems from the particle filtering mechanism’s ability to rapidly update belief distributions after each sample, achieving convergence without

overfitting to large training sets.

##### 4.7.2 Impact of Context Window Length on BCL Performance

Table 5: F1 scores (%) for different context lengths (in sentences) during the observation step

Context Length	CoNLL03	GENIA	ACE05
1	63.37	45.20	38.15
2	68.25	48.60	42.30
4	72.04	51.80	46.95
6	71.10	50.90	46.50
8	71.04	50.20	46.20
10	60.20	47.30	39.80
12	59.85	46.80	39.20
14	60.15	47.10	39.60

The context window length (also called batch size) determines how much information our method considers when computing posterior probabilities at each observation step. Table 5 shows that on Qwen-2.5-7B, performance improves significantly from single-sentence to five-sentence contexts due to smoother joint likelihood functions that provide more stable gradients. However, performance declines beyond 9 sentences as excessive observations lead to overly averaged particle weights, preventing effective updates.

## 5 Conclusion

In this paper, we propose BCL, a general framework that can efficiently traverse training sets and rapidly extract specific relationships from training data using particle-based methods. BCL is the first optimization framework using Bayesian inference for IE tasks. We have conducted extensive and comprehensive experiments to thoroughly demonstrate the effectiveness of our approach. Additionally, we acknowledge the limitations of our method, such as the generation of a large number of particles to achieve rapid convergence, which increases inference burden and computational costs in practice. Therefore, our future research will focus on improving particle utilization efficiency to further enhance the overall system performance.

## Acknowledgement

This work was supported in part by the National Key R&D Program under grants 2025ZD1502903 and 2024YFC3308101.

## Limitations

Our work has several limitations. First, while BCL maintains reasonable inference efficiency, the optimization phase requires  $O(K \times M)$  iterations, where  $K$  denotes the number of particles and  $M$  denotes the number of training samples. Each iteration involves CPU-intensive numerical computations including weight updates and resampling (detailed analysis in Appendix E.3). This makes the method most suitable for scenarios where optimization costs can be amortized across multiple deployments, rather than one-time or frequently-updated applications.

Second, the particle filtering mechanism may bias optimization toward frequent patterns in the training data, potentially affecting rare entity types. However, our empirical analysis (Appendix D) shows that BCL is largely robust to such frequency imbalance, with only marginal performance differences across frequency strata.

## References

- Karl Johan Åström and Richard Murray. 2021. *Feedback systems: an introduction for scientists and engineers*. Princeton university press.
- Mark A Beaumont, Wenyang Zhang, and David J Balding. 2002. Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Chengkun Cai, Haoliang Liu, Xu Zhao, Zhongyu Jiang, Tianfang Zhang, Zongkai Wu, John Lee, Jenq-Neng Hwang, and Lei Li. 2025a. Bayesian optimization for controlled image editing via llms. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Chengkun Cai, Xu Zhao, Haoliang Liu, Zhongyu Jiang, Tianfang Zhang, Zongkai Wu, Jenq-Neng Hwang, and Lei Li. 2025b. The role of deductive and inductive reasoning in large language models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Linbo Cao and Jinman Zhao. 2025. [Pretraining on the test set is no longer all you need: A debate-driven approach to QA benchmarks](#). In *Second Conference on Language Modeling*.
- Chao Chen, Pengfei Luo, Changkai Feng, Tian Wu, Wenbin Jiang, and Tong Xu. 2025. [Maps: A multi-task framework with anchor point sampling for zero-shot entity linking](#). *DATA INTELLIGENCE*, 7(4):1085–1107.
- Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, and 1 others. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.
- Arnaud Doucet, Nando De Freitas, Neil James Gordon, and 1 others. 2001. *Sequential Monte Carlo methods in practice*, volume 1. Springer.
- Peter I Frazier. 2018. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*.
- Neil J Gordon, David J Salmond, and Adrian FM Smith. 1993. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET.
- Yunchuan Guan, Yu Liu, Ke Zhou, Zhiqi Shen, Jenq-Neng Hwang, Serge Belongie, and Lei Li. 2025. Is meta-learning out? rethinking unsupervised few-shot classification with limited entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4188–4197.
- Yunchuan et al. Guan. 2025. Learning an efficient optimizer via hybrid-policy sub-trajectory balance. *arXiv*.
- Shizhou Huang, Bo Xu, Yang Yu, Changqun Li, and Xin Alex Lin. 2025. Guidener: Annotation guidelines are better than examples for in-context named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24159–24166.
- Yuelyu Ji, Zhuochun Li, Rui Meng, and Daqing He. 2025. [Reason-to-rank: Distilling direct and comparative reasoning from large language models for document reranking](#). In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '25*, page 2320–2329, New York, NY, USA. Association for Computing Machinery.
- Yuelyu Ji, Zhuochun Li, Rui Meng, and Daqing He. 2026. Retrieval–reasoning processes for multi-hop question answering: A four-axis design framework and empirical trends. *arXiv preprint arXiv:2601.00536*.
- J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun’ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl\_1):i180–i182.
- Tian Lan, Jinyuan Xu, Xue He, Jenq-Neng Hwang, and Lei Li. 2025. Attention consistency for llms explanation. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 1736–1750.

- Guozheng Li, Peng Wang, Wenjun Ke, Yikai Guo, Ke Ji, Ziyu Shang, Jiajun Liu, and Zijie Xu. 2024a. Recall, retrieve and reason: towards better in-context relation extraction. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 6368–6376.
- Lei Li, Sen Jia, and Jenq-Neng Hwang. 2026. Multiple human motion understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 6297–6305.
- Lei Li, Sen Jia, Jianhao Wang, Zhongyu Jiang, Feng Zhou, Ju Dai, Tianfang Zhang, Zongkai Wu, and Jenq-Neng Hwang. 2025. Human motion instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuan-Jing Huang, and Xipeng Qiu. 2023a. Codeie: Large code generation models are better few-shot information extractors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15339–15353.
- Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023b. Unified demonstration retriever for in-context learning. *arXiv preprint arXiv:2305.04320*.
- Yinghao Li, Rampi Ramprasad, and Chao Zhang. 2024b. A simple but effective approach to improve structured language model output for information extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5133–5148.
- Jiaxin Liu. 2026. [Discovering what you can control: Interventional boundary discovery for reinforcement learning](#). *Preprint*, arXiv:2603.18257.
- Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Zhao. 2024. [Dr.icl: Demonstration-retrieved in-context learning](#). *DATA INTELLIGENCE*, 6(4):909–922.
- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.
- Ying Mo, Jiahao Liu, Jian Yang, Qifan Wang, Shun Zhang, Jingang Wang, and Zhoujun Li. 2024. C-icl: Contrastive in-context learning for information extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10099–10114.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*.
- Biqing Qi, Zhouyi Qian, Yiang Luo, Junqi Gao, Dong Li, Kaiyan Zhang, and Bowen Zhou. 2024. Evolution of thought: Diverse and high-quality reasoning via multi-objective optimization. *arXiv preprint arXiv:2412.07779*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.
- Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 148–163. Springer.
- Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks.
- Jingzhe Shi, Qinwei Ma, Hongyi Liu, Hang Zhao, Jenq-Neng Hwang, and Lei Li. 2026. Intrinsic entropy of context length scaling in llms. In *International Conference on Learning Representations (ICLR)*.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. [Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition](#). In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.
- Somin Wadhwa, Silvio Amir, and Byron C Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589.
- Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. (*No Title*).
- Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. Gpt-re: In-context learning for relation extraction using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3534–3547.
- Zihang Wang, Ye Liang, Wenwei Sun, Qicong Lin, Chao Xu, and Yong Zhang. 2025. [A novel feature selection framework based on large language models](#). *DATA INTELLIGENCE*, 7(4):1016–1034.
- Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and 1 others. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, and 1 others. 2023. Chatie: Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Tianyi Xu, Jiaxin Liu, Nicholas Mattei, and Zizhan Zheng. 2026. Fair algorithms with probing for multi-agent multi-armed bandits. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(32):27332–27340.

Ziyu Yao, Xuxin Cheng, Zhiqi Huang, and Lei Li. 2025. Countllm: Towards generalizable repetitive action counting via large language model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Qi Zhang, Zhijia Chen, Huitong Pan, Cornelia Caragea, Longin Jan Latecki, and Eduard Dragut. 2024. Scier: An entity and relation extraction dataset for datasets, methods, and tasks in scientific documents. *arXiv preprint arXiv:2410.21155*.

Jinman Zhao, Erxue Min, Hui Wu, Ziheng Li, Zexu Sun, Hengyi Cai, Shuaiqiang Wang, Xu Chen, and Gerald Penn. 2026. Beyond step pruning: Information theory based step-level optimization for self-refining large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(41):34941–34949.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. In *The eleventh international conference on learning representations*.

## A Dataset Statistics

We provide detailed statistics for all six datasets used in our experiments in Table 6.

## B Case Study

We present a case study using a CoNLL-2003 sentence to illustrate our three-stage workflow: particle generator, posterior probability calculator, and resampler. Here,  $x$  represents label sequences,  $N$  denotes the number of generation rules, and  $Y$  represents input text. It is worth noting that the following case studies present simplified examples to

Table 6: Statistics of the datasets used in our experiments.  $|\text{Ents}|$  and  $|\text{Rels}|$  denote the number of entity types and relation types.  $\#\text{Train}$ ,  $\#\text{Val}$  and  $\#\text{Test}$  denote the sample number in each split.

Dataset	$ \text{Ents} $	$ \text{Rels} $	$\#\text{Train}$	$\#\text{Val}$	$\#\text{Test}$
<i>Named Entity Recognition</i>					
CoNLL03	4	-	14,041	3,250	3,453
ACE05	7	-	6,202	745	812
GENIA	5	-	15,023	1,669	1,854
<i>Relation Extraction</i>					
NYT	-	24	56,195	5,000	5,000
CoNLL04	4	5	922	231	288
SciERC	6	7	1,861	275	551

illustrate our framework’s workflow. All experiments follow the technical specifications detailed in Section 3.

### B.1 Case Study 1: Particle Generator

Particle generation serves as the initial step in our framework to obtain initial particles and compute their corresponding prior probabilities. We demonstrate this process using the label "organization" as an example. As shown in Figure 5, our prompt template instructs the model to act as a subcategory generation expert, decomposing broad entity labels into semantically distinct subcategories with associated probability weights. For the "organization" label, the system generates diverse subcategories such as "sports team", "city", "company", "name", and "financial institution". Each subcategory represents a different semantic interpretation of how organizations might appear in text, with the probability weights serving as prior probabilities that reflect their typical occurrence frequency.

### B.2 Case Study 2: Posterior Probability Calculator

Following particle generation, we compute posterior probabilities by evaluating how well each generated rule performs on the actual classification task. Figure 6 illustrates this process using the input text "EU rejects German call to boycott British lamb." The system applies the previously generated subcategory rules to identify entities in the given text. The likelihood of each particle is calculated based on the model’s ability to correctly identify and classify entities according to the generated rules. The posterior probability calculation applies Bayes’ theorem to update particle weights based on how well each rule matches the

<p><b>Prompt Template:</b>  Create subcategories for: X  Format: {"X": {subcategory}}</p>
<p><b>Input:</b>  X:Person, Organization, Location, ...</p>
<p><b>Output Rules:</b>  "organization": ["sports tesm", "city", "company", "name", "financial institution",...]  "person": ["name", "full name", "athlete", "individual", "public figure"],...</p>

Figure 5: Case study showing the prompt template for subcategory generation. The template guides the model to generate semantic subcategories for named entity labels, demonstrated with organization and person entity types.

observed entity patterns in the input text. For instance, the "organization" subcategories show varying performance scores: "sports team: 0.9277", "city: 0.7073", "company: 0.4751", demonstrating how different semantic interpretations receive different posterior weights based on their effectiveness. This likelihood-based evaluation ensures that particles with better classification performance receive higher posterior probabilities, enabling the system to focus on the most promising labeling hypotheses for subsequent resampling.

### B.3 Case Study 3: Resampler

The final stage employs an LLM-based resampling mechanism that performs context-aware particle mutation. As shown in Figure 7, given the input text "He said further scientific study was required and if it was found that action was needed it should be taken by the European Union," the system generates contextually relevant subcategories. The resampling process works as follows: the LLM analyzes the specific context and mutates the original particles to better fit the observed text patterns. For the "organization" label, the system generates context-specific subcategories such as "educational institution," "non-governmental organization," "research institution," and "financial institution," which are more relevant to the scientific and policy context of the input. Prior probabilities are computed to indicate higher prior probability. For instance, "educational institution: 0.340369" and "non-governmental organization: 0.297805" receive relatively high priors due to their contextual relevance. This perplexity-based weighting ensures that contextually appropriate mutations are favored

during the resampling process.

## C Ablation on Semantic Decomposition

To further validate the necessity of semantically coherent subcategories, we design a controlled ablation experiment that breaks the semantic alignment between subcategories and their corresponding entity labels.

We take the optimized subcategory set learned by BCL and randomly reassign subcategories to different entity labels. For example, subcategories such as "company" or "event" may be assigned to the "Person" label instead of semantically consistent categories like "athlete" or "politician". Importantly, the subcategory pool remains unchanged, ensuring that the only difference lies in the loss of semantic alignment.

Table 7 shows the results across representative datasets and models. Breaking semantic coherence leads to consistent performance degradation across all settings.

Dataset	Model	Semantic	Shuffled	$\Delta$
CoNLL03	Qwen-7B	72.83	67.20	-5.63
CoNLL03	Llama-8B	69.14	56.90	-12.24
ACE05	Qwen-7B	46.94	38.90	-8.04
ACE05	Llama-8B	53.10	49.50	-3.60

Table 7: Effect of semantic decomposition.

On average, shuffling subcategories results in a 7.38 F1 point drop, demonstrating that semantic coherence plays a crucial role in enabling effective optimization. Meanwhile, the framework still maintains reasonable performance, suggesting that BCL exhibits graceful degradation rather than catastrophic failure.

<p><b>Prompt Template:</b>  Classify entities in the text into predefined categories.  Label: Person, Organization, Location, ...  Input: Y  Output: Structured entity-category pairs</p>
<p><b>Input Text:</b>  Eu rejects German call to boycott British lamb.</p>
<p><b>Model Output Rules:</b>  "organization": ["sports team", "city", "company", "name", "financial institution",...]  "person": ["name", "full name", "athlete", "individual", "public figure",...]</p>
<p><b>Model Output Rules and Calculated Posterior Probability:</b>  "organization": ["sports team: 0.9277", "city: 0.7073", "company: 0.4751", "name: 0.7891", "financial institution: 0.6779",...]  "person": ["name: 0.6850", "full name: 0.5437", "athlete: 0.1327", "individual: 0.6227", "public figure: 0.7104",...]</p>

Figure 6: Case study demonstrating posterior probability calculation through likelihood estimation. The model evaluates generated rules against input text to compute particle posterior probabilities based on classification accuracy.

Model	Dataset	Top 30%	Mid 30%	Bottom 40%
Llama-3.1-8B	ACE05	53.89	55.25	51.94
Llama-3.1-8B	CoNLL2003	69.43	69.64	68.23
Qwen2.5-7B	ACE05	46.62	47.61	47.03
Qwen2.5-7B	CoNLL2003	72.39	73.31	73.67

Table 8: Frequency-stratified F1 performance of BCL.

## D Frequency-Stratified Analysis on Long-Tail Entities

To further investigate the potential long-tail effect discussed in Section X, we conduct a frequency-stratified evaluation to quantify model performance across entities with different training frequencies.

We perform experiments on two models (Llama-3.1-8B and Qwen2.5-7B) and two datasets (ACE05 and CoNLL2003). Entities in the test set are grouped into three tiers based on their frequency in the training set: Top 30% (frequent), Mid 30% (medium-frequency), and Bottom 40% (rare). We report F1 scores for each group separately.

Table 8 summarizes the results. Overall, BCL demonstrates strong robustness to frequency imbalance. The performance gap between frequent and rare entities is generally small (within 1–2 F1 points), and in some cases, rare entities even achieve comparable or slightly better performance.

These results suggest that, although the particle filtering mechanism theoretically emphasizes frequent patterns, its impact on rare entity perfor-

mance is limited in practice. We hypothesize that the Bayesian aggregation process helps mitigate overfitting to frequent patterns by maintaining diverse hypothesis particles.

## E Implementation Details

### E.1 Prior and Likelihood Selection in Bayesian Filtering

In our Bayesian filtering framework, particle weights are updated through the combination of prior beliefs and observed evidence. We use perplexity-based scores as the prior and task-specific confidence as the likelihood, based on the following considerations:

**(1) Rationality of the Prior:** Perplexity reflects the degree of consistency between generated rules and the LLM’s internal language distribution. Lower perplexity indicates that the rule better aligns with the model’s “linguistic intuition,” and as a prior assumption, such rules are more likely to yield high-quality extraction results. This provides a reasonable inductive bias: before observing any

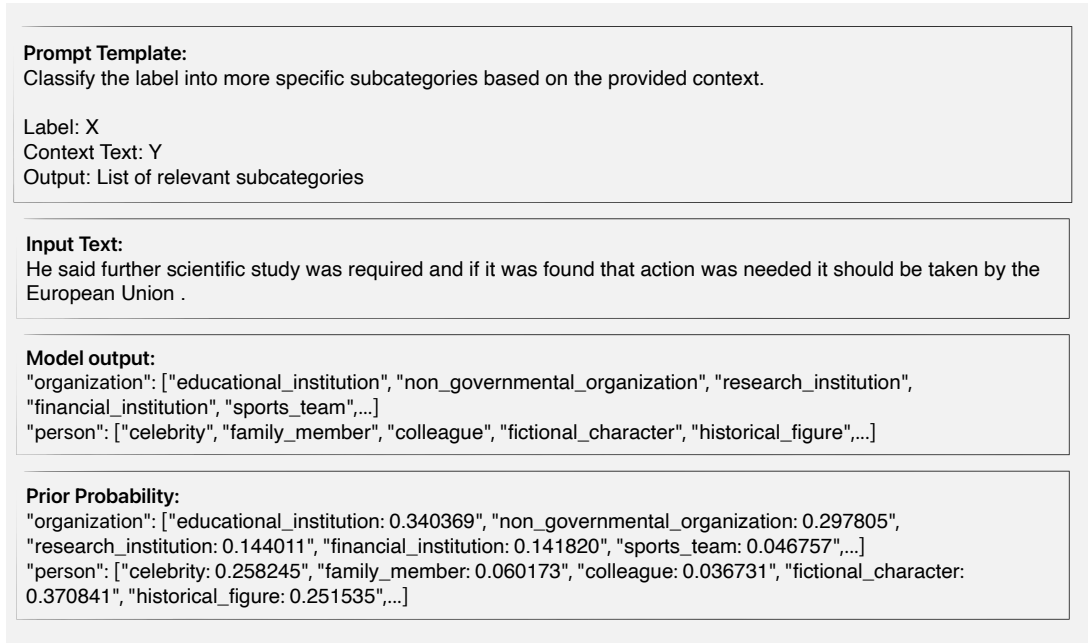


Figure 7: Case study demonstrating the resampling stage with particle mutation.

task-specific performance, we favor rules that are linguistically coherent and natural according to the model’s pre-trained knowledge.

**(2) Rationality of the Likelihood:** The IE confidence score  $L_c$  (Equation 12) measures how well a rule performs on actual extraction tasks. Unlike perplexity, which only captures linguistic properties, the confidence score directly evaluates the rule’s effectiveness in guiding the LLM to generate correct entity labels. This task-specific measurement serves as the likelihood function  $p(D|\theta)$ , providing empirical evidence to update our beliefs about rule quality based on observed extraction performance.

**(3) Separation of Prior and Likelihood:** In standard Bayesian updating, the prior  $p(\theta)$  and likelihood  $p(D|\theta)$  serve distinct roles. Similarly, in our framework:

- **Perplexity (Prior):** Encodes the inductive bias that “rules should be linguistically fluent and coherent”
- **IE Confidence (Likelihood):** Provides observational evidence through actual task performance

This separation allows newly generated particles to carry reasonable initial beliefs into the evaluation stage, rather than using uninformative uniform priors. The Bayesian update mechanism (Equation 13)

then combines these two sources of information:

$$\underbrace{w_i^{(t)}}_{\text{posterior}} \propto \underbrace{w_i^{(t-1)}}_{\text{prior}} \cdot \underbrace{\exp(\beta \cdot y_i^{(t)})}_{\text{likelihood}} \quad (18)$$

**(4) Robustness via Filtering Dynamics:** Even if perplexity imperfectly approximates rule quality, the particle filtering framework provides inherent robustness through its denoising mechanism. Particles with misleading priors (low perplexity but poor IE performance) receive low likelihood scores, causing their weights to exponentially decay through multiplicative updates and be eliminated during resampling. This mechanism ensures that the final particle distribution is dominated by task performance rather than initial priors.

## E.2 Evaluation Metric.

Following previous work (Huang et al., 2025), we adopt entity-level F1 scoring for both NER and RE tasks. For NER, a predicted entity is considered correct only when both its boundary (character span) and type label exactly match the ground truth annotation. For RE, both entity mentions and their relation type must match exactly. Precision, recall, and F1 score are computed at the entity/relation level across the entire test set.

### E.3 Computational Complexity.

Let  $K$  denote the number of particles per label and  $M$  denote the number of training samples used for optimization. Our method processes batches sequentially through iterative filtering, requiring  $O(K \times M)$  LLM inference calls in total. In contrast, methods like GuideNER (Huang et al., 2025) require  $O(N)$  calls to traverse the complete training set of size  $N$ . Since we operate on  $M \ll N$  samples (e.g.,  $M \approx 0.03N$  as shown in Section 4.7.1), our method achieves significant efficiency gains. The weight update (Equation 14) and resampling (Equation 17) operations involve only  $O(K)$  numerical computations—probability calculations, softmax normalization, sorting, and random sampling—which are negligible compared to LLM inference time. Therefore, the computational cost is dominated by LLM inference, and our approach is substantially more efficient than full-dataset traversal methods.

### E.4 Hyperparameters.

Key hyperparameters are set as follows: number of particles per label  $N = 10$  (empirically optimal range: 1–20), selection pressure  $\beta = 2.0$  (selected via grid search over  $\{1, 2, 5, 10\}$  across multiple datasets), observation batch size of 4 sentences (analyzed in Section 4.7.2), and retention ratio of 50% for resampling (balancing exploitation and exploration). All LLM inference is performed with temperature=0.0 and random seed=42 for reproducibility. Convergence is reached when validation F1 score plateaus for 3 consecutive iterations.

## F Check List

This appendix provides additional details required by the ACL Responsible NLP Research checklist.

### F.1 Data Licenses and Terms of Use (B2)

We provide license information for all datasets used in our experiments:

**Model Licenses.** The language models used in our experiments are distributed under the following licenses:

- **Qwen2.5-3B/7B:** Apache 2.0 License (Qwen Team, Alibaba)
- **Llama-3.1-8B:** Llama 3.1 Community License (Meta)
- **Pixtral-12B:** Apache 2.0 License (Mistral AI)

Dataset	License/Terms	Access
<i>Named Entity Recognition</i>		
CoNLL-2003	Research use only	Public
ACE 2005	LDC User Agreement	LDC
GENIA	GENIA Project License	Public
<i>Relation Extraction</i>		
NYT	LDC (derived)	Public
CoNLL04	Research use only	Public
SciERC	CC BY 4.0	Public

Table 9: License information for all datasets used in experiments.

**Code Availability.** Our implementation will be released under the MIT License upon acceptance. The codebase includes all scripts for data preprocessing, particle filtering optimization, and evaluation.

### F.2 Intended Use and Consistency (B3)

All datasets were used in accordance with their intended purposes:

- **CoNLL-2003:** Originally created for the CoNLL-2003 shared task on language-independent named entity recognition. We use it for NER evaluation as intended.
- **ACE 2005:** Developed for entity, relation, and event extraction research. We use the entity annotations for NER evaluation.
- **GENIA:** Created for biomedical text mining research. We use it for biomedical NER evaluation as intended.
- **NYT:** Derived from New York Times articles for relation extraction research. We use it for RE evaluation.
- **CoNLL04:** Designed for joint entity and relation extraction. We use it for RE evaluation.
- **SciERC:** Created for information extraction from scientific papers. We use it for scientific RE evaluation as intended.

Our derived artifacts are intended solely for research purposes in information extraction and should not be used for commercial applications without appropriate licensing.

### F.3 Privacy and Offensive Content (B4)

We did not perform additional anonymization as the original dataset creators have already addressed privacy considerations in their data collection and release procedures.

## F.4 Artifact Documentation (B5)

**Dataset Coverage.** Table 10 provides detailed documentation of the datasets used:

Dataset	Domain	Language	Text Source
CoNLL-2003	News	English	Reuters
ACE 2005	News/Conv.	English	Various news
GENIA	Biomedical	English	PubMed abstracts
NYT	News	English	New York Times
CoNLL04	General	English	News articles
SciERC	Scientific	English	AI paper abstracts

Table 10: Domain, language, and source documentation for all datasets.

### Entity and Relation Types.

- **CoNLL-2003 (4 types):** PER, LOC, ORG, MISC
- **ACE 2005 (7 types):** Person, Organization, Location, Facility, Weapon, Vehicle, GPE
- **GENIA (5 types):** DNA, RNA, Protein, Cell Line, Cell Type
- **NYT (24 relation types):** Including location-related, person-related, and organization-related relations
- **CoNLL04 (5 relation types):** Located-In, Work-For, OrgBased-In, Live-In, Kill
- **SciERC (7 relation types):** Used-for, Feature-of, Part-of, Compare, Hyponym-of, Evaluate-for, Conjunction

## F.5 Package Parameters (C4)

### Software Dependencies.

- Python 3.10.12
- PyTorch 2.6.0
- Transformers 4.51.3
- NumPy 1.24.3
- scikit-learn 1.3.0 (for evaluation metrics)

**Evaluation Implementation.** We use the `seqeval` library (v1.2.2) for NER evaluation with the following settings:

- `mode='strict'`: Exact boundary and type matching required
- `scheme=IOB2`: Using IOB2 tagging scheme

For RE evaluation, we implement custom evaluation following prior work (Riedel et al., 2010):

- A relation is correct if both entity mentions and relation type match exactly
- Precision, recall, and F1 are computed at the relation-tuple level

### Text Preprocessing.

- Tokenization: Model-specific tokenizers from HuggingFace
- No additional preprocessing (lowercasing, stemming) applied
- Maximum sequence length: 512 tokens (truncation applied if exceeded)

## F.6 Use of AI Assistants (E1)

In this paper, we employed Large Language Models (LLMs) as core components of our proposed BCL framework in three specific stages. First, we utilized LLMs for rule extraction and particle generation to create initial subcategory patterns from training data, ensuring systematic rule discovery. Second, we leveraged LLMs for performance evaluation through in-context learning inference, where models assess rule effectiveness on validation datasets. Third, we employed LLMs for rule mutation and resampling to generate semantically diverse rule variants during the optimization process.

Additionally, we used LLMs to assist with writing and language improvement throughout the manuscript preparation process. This included grammar checking, sentence structure optimization, and clarity enhancement to improve the overall readability of our work. However, all core research ideas, methodological contributions, experimental designs, and conclusions remain entirely our own intellectual work.

## F.7 Broader Impact Statement

**Positive Impacts.** BCL advances information extraction technology with several benefits:

- **Accessibility:** By improving ICL performance on smaller models (3B-12B parameters), BCL democratizes access to effective IE systems for researchers and practitioners with limited computational resources.

- **Efficiency:** The data-efficient optimization (converging with 3-5% of training data) reduces the annotation burden and computational costs.
- **Generalizability:** The framework's applicability to both NER and RE tasks provides a unified approach to diverse IE challenges.

**Potential Negative Impacts.**

- **Automation of Information Extraction:** While beneficial for legitimate applications, improved IE could potentially be misused for unauthorized data harvesting or surveillance.
- **Environmental Impact:** Although more efficient than full fine-tuning, LLM-based IE still requires significant computational resources with associated carbon emissions.

**Mitigation Strategies.** We encourage users to:

1. Apply BCL only to data they have legal rights to process
2. Implement human oversight in high-stakes applications
3. Consider the environmental impact and use appropriately-sized models for their needs