MEIC-DT: Memory-Efficient Incremental Clustering for Long-Text Coreference Resolution with Dual-Threshold Constraints

Kangyang Luo[♠], Shuzheng Si[♠], Yuzhuo Bai[♠], Cheng Gao[♠], Zhitong Wang[♠]
Cheng Huang[♠], Yingli Shen[♠], Yufeng Han[♠], Wenhao Li[♠], Cunliang Kong[♠]
Maosong Sun[*♠♣★]
[♠]Department of Computer Science and Technology, Tsinghua University
[♣]Institute for AI, Tsinghua University
[★]Jiangsu Collaborative Innovation Center for Language Ability

Abstract

In the era of large language models (LLMs), supervised neural methods remain the state-of-the-art (SOTA) for Coreference Resolution. Yet, their full potential is underexplored, particularly in incremental clustering, which faces the critical challenge of balancing efficiency with performance for long texts. To address the limitation, we propose **MEIC-DT**, a novel dual-threshold, memory-efficient incremental clustering approach based on a lightweight Transformer. MEIC-DT features a dual-threshold constraint mechanism designed to precisely control the Transformer's input scale within a predefined memory budget. This mechanism incorporates two key components: a Statistics-Aware Eviction Strategy (**SAES**) and an Internal Regularization Policy (**IRP**). The SAES utilizes distinct statistical profiles from the training and inference phases for intelligent cache management. The IRP strategically condenses clusters by selecting the most representative mentions, thereby preserving semantic integrity. Extensive experiments on common benchmarks demonstrate that MEIC-DT achieves highly competitive coreference performance under stringent memory constraints.

1 Introduction

Coreference Resolution (CR) is a fundamental task in Natural Language Processing (NLP), which aims to identify different text spans (i.e., mentions) within a document that refer to the same entity (Karttunen, 1976; Lee et al., 2017, 2018). It plays a critical role in a variety of downstream applications, such as text summarization (Liu et al., 2024b), knowledge graph construction (Pan et al., 2024; Chen et al., 2025), question answering(Pan et al., 2024; Jang et al., 2025), and named entity recognition (Shang et al., 2025). Current mainstream CR methods can fall into three types: *supervised neural methods* (Xu and Choi, 2020; Wu et al., 2020; Kirstain et al., 2021; Lai et al., 2022; Otmazgin et al., 2022a,b; Guo et al., 2023; Martinelli et al., 2024), *generative methods* (Liu et al., 2022; Bohnet et al., 2023; Zhang et al., 2023), and *large language model-based methods* (Le and Ritter, 2023; Gan et al., 2024; Zhu et al., 2025). Among these, supervised neural methods demonstrate significant advantages in terms of training cost, inference efficiency, while achieving competitive performance. However, their potential, particularly for challenging scenarios like long-text processing, has not yet been fully explored.

Within *supervised neural methods*, incremental clustering—a strategy inspired by human incremental cognitive processing (Altmann and Steedman, 1988)—is widely adopted for long-text scenarios. However, despite significant progress in CR, existing methods (Xia et al., 2020; Toshniwal et al., 2020; Guo et al., 2023; Martinelli et al., 2024) still face a critical challenge in balancing efficiency and performance during mention clustering. For example, conventional incremental clustering approaches (Xia et al., 2020; Toshniwal et al., 2020; Guo et al., 2023) typically employ a fixed-size cache and a linear classifier to associate candidate mentions with coreference clusters (each represented as a single vector) tracked by the cache. These approaches decide whether a mention should join an existing cluster or form a new one. Upon reaching capacity, a predefined rule-based eviction strategy—LRU (*Least Recently Used*) or a dual-cache mechanism combining LRU and LFU (*Least Frequently Used*)—removes certain entities to update the cache dynamically. While computationally efficient, such approaches often fail to capture the complex nonlinear dependencies between mentions and clusters, and result in limited semantic representations due to excessive intra-cluster compression. Moreover, these strategies either rely on a single heuristic (e.g., LRU) that may discard essential clusters, or employ a dual-cache mechanism

[*]Corresponding author

Findings of the Association for Computational Linguistics: ACL 2026, pages 17787–17804

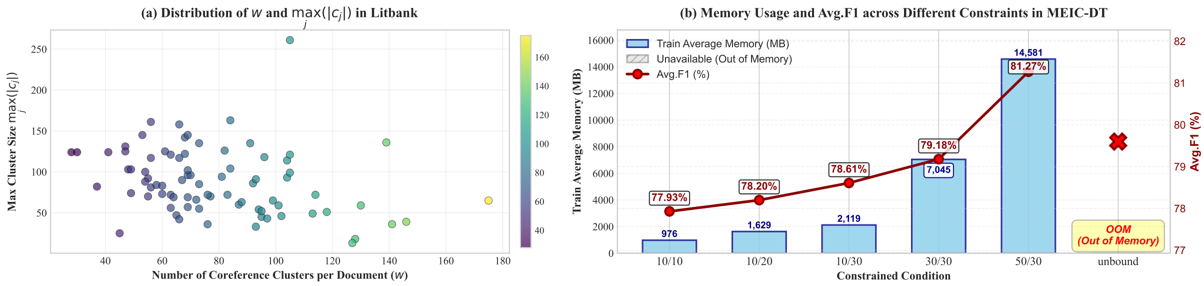

Figure 1: Analysis of motivations on the LitBank training corpus. Notably, the "unbound" condition failed due to out-of-memory (OOM) errors under our memory constraints.

that adds to the complexity of cache management. Crucially, the same eviction strategy is used across both training and inference. This does not account for the inherent differences in information availability between the two phases and thus hinders the model's overall effectiveness.

Recently, Martinelli et al. (2024) introduced a lightweight Transformer model and achieved SOTA performance. Its input is formed by concatenating the representation of a candidate mention with representations of all mentions from historical coreference clusters, effectively modeling the global intra-cluster structure and significantly improving clustering accuracy. However, a major drawback is that the input size grows polynomially with the number of clusters and the number of mentions per cluster, causing a surge in memory usage and computational latency. This severely limits the model's practicality and scalability under typical academic computational resources. ***These considerations motivate our investigation into pushing the boundaries of supervised neural models in the era of LLMs.***

To tackle the above issues, we propose **MEIC-DT**, a novel memory-efficient incremental clustering approach with dual-threshold constraints based on a lightweight Transformer, which aims to achieve competitive coreference resolution performance while adhering to a strict, predefined memory budget. MEIC-DT formulates a ***dual-threshold constraint mechanism*** that regulates the Transformer's input scale along two key dimensions: first, by leveraging distinctive information available during training and inference, we design a ***statistics-aware eviction strategy*** to preserve crucial coreference clusters within the constrained cache space; second, we introduce an ***internal regularization policy*** that limits the number of mentions per cluster, enabling the selection of only the most representative mentions for computation under tight memory conditions. To sum up, we highlight our contributions as follows:

- We propose a novel approach MEIC-DT, which reduces the memory and computational burden of Transformer-based coreference clustering while maintaining competitive performance.
- We design a dual-threshold constraint mechanism, integrating a statistics-aware eviction strategy for phase-adaptive cache management and an internal regularization policy for cluster condensation, to ensure robust and efficient operation under stringent memory limits.
- Extensive experiments on several commonly used CR benchmarks (i.e., OntoNotes, LitBank and WikiCoref) show that MEIC-DT achieves competitive coreference performance in memory-constrained scenarios.

2 Preliminaries

Notations. Given a long text $D = [t_1, t_2, \ldots, t_n]$ composed of n tokens, let $\mathbf{H} = [\mathbf{h}_1, \cdots, \mathbf{h}_n] \in \mathbb{R}^{n \times d_h}$ denote the hidden representation of D from a text encoder, where d_h is the hidden dimension. During mention clustering, the set of candidate mentions output by a mention extractor is defined as $M = [m_1, m_2, \cdots, m_r]$, where each mention m_i is represented by the start and end tokens $(t_{i,s}, t_{i,e})$. Historical coreference clusters (i.e., entities) are denoted as $\{c_j\}_{j=1}^{w} = \{[m^f_{j,1}, \cdots, m^g_{j,|c_j|}]\}_{j=1}^{w}$, where f and g are indices of mentions in M with $f \leq g$, and w is the total number of clusters.

Transformer-based incremental model. To our knowledge, Martinelli et al. (2024) were the first to

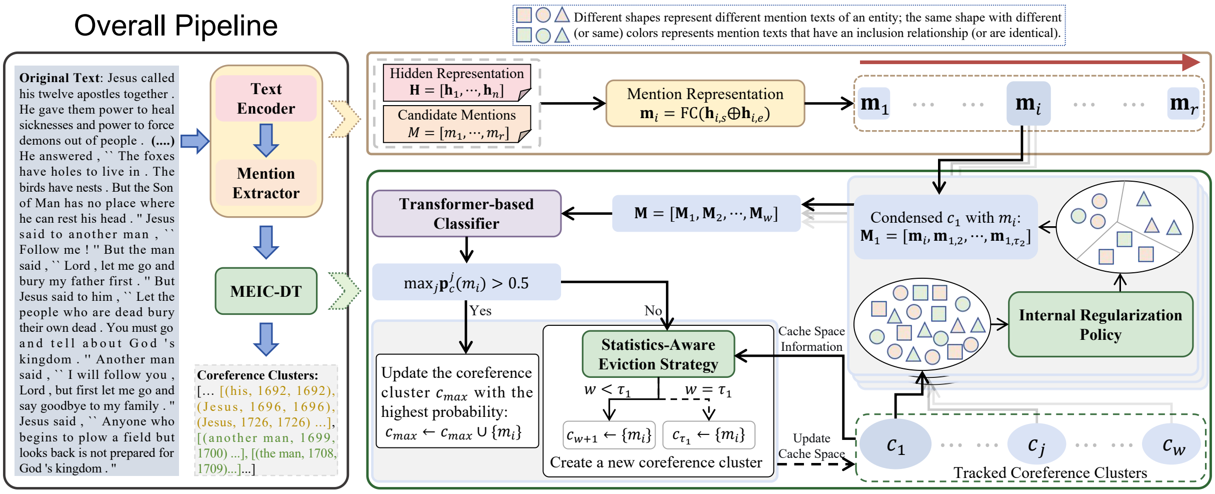

Figure 2: The MEIC-DT Coreference Resolution pipeline. The core innovation is a dual-threshold constraint mechanism—comprising a *statistics-aware eviction strategy* and an *internal regularization policy*—that enables high-performance, memory-efficient incremental clustering.

employ a lightweight Transformer for mention clustering. Their model achieves SOTA performance on long texts by computing clustering probabilities between candidate mentions and existing clusters, using all mentions in them as input.

Specifically, the representation of a candidate mention m_i is formed by concatenating the hidden states of its start and end tokens, followed by a fully connected (FC) layer, i.e., $\mathbf{m}_i = \text{FC}(\mathbf{h}_{i,s} \oplus \mathbf{h}_{i,e}) \in \mathbb{R}^{d_h}$, where $\oplus$ denotes concatenation operator. For historical clusters $\{c_j\}_{j=1}^{w}$ (where $f \leq g < i$), the representations of all mentions in each cluster are combined with $\mathbf{m}_i$ as follows:

$$\mathbf{M}_j = [\mathbf{m}_i, \mathbf{m}_{j,1}^{f}, \cdots, \mathbf{m}_{j,|c_j|}^{g}],$$
$$\mathbf{M} = [\mathbf{M}_1; \cdots; \mathbf{M}_w], \qquad (1)$$

where $\mathbf{M} \in \mathbb{R}^{w \times (\max_j(|c_j|)+1) \times d_h}$. Notably, clusters with fewer than $\max_j(|c_j|)$ mentions are padded with the [PAD] token's embedding. A Transformer (TF) encoder then processes $\mathbf{M}$, and the output of the [CLS] token is used:

$$\mathbf{T}_i^{[CLS]} = \text{TF}(\mathbf{M}) \in \mathbb{R}^{w \times d_h}. \qquad (2)$$

Finally, an MLP classifier followed by a sigmoid function σ computes the clustering probabilities:

$$\mathbf{p}_c(m_i) = \sigma(\text{MLP}(\mathbf{T}_i^{[CLS]})) \in \mathbb{R}^{w \times 1}. \qquad (3)$$

The mention m_i is assigned to the cluster with the highest probability if $\max_j \mathbf{p}_c^j(m_i) > 0.5$; otherwise, a new cluster is created. The process is initialized with $c_1 = [m_1]$.

3 Methodology

3.1 Analytical Motivation

Although Eq. (2) can model the global information of historical coreference clusters, the dimension of $\mathbf{M} \in \mathbb{R}^{w \times (\max_j(|c_j|)+1) \times d_h}$ expands rapidly as w and $\max_j(|c_j|)$ increase. This leads to substantial growth in memory usage during training, severely limiting the model's practicality for long texts. To illustrate this issue, Fig. 1(a) shows the distributions of w and $\max_j(|c_j|)$ in the LitBank training corpus. It shows that many samples have large w or $\max_j(|c_j|)$ values, directly inflating the dimension of $\mathbf{M}$. Under computationally constrained settings, even using a lightweight Transformer model entails a non-negligible overhead, as shown in Fig. 1(b). See Section 4 for more details.

To address this, the proposed MEIC-DT approach employs a dual-threshold mechanism to control the input size of the lightweight Transformer by simultaneously limiting the number of historical coreference clusters and the number of mentions per cluster. Thereafter, we detail the design of MEIC-DT, including *Statistics-Aware Eviction Strategy* and *Internal Regularization Policy*.

3.2 Statistics-Aware Eviction Strategy

To retain crucial coreference clusters in a fixed-size cache, we propose a Statistics-Aware Eviction Strategy (**SAES**). SAES leverages the distinct information available in each phase: global statistics during training and real-time state during inference. This allows us to design differentiated, phase-specific eviction criteria. Our strategy not only bounds computational cost by limiting the cache to at most τ_1 clusters in the cache, where $\tau_1 > 0$ is a predefined integer threshold, but also improves the cache hit rate of target mentions.

Specifically, ***during training***, where ground-truth coreference cluster annotations are available, we construct a ranking score for each tracked cluster c_j ($j \in [1, \tau_1]$) based on its number of remaining mentions rm_j and the number of steps since its last access lru_j:

$$SAES_{\text{train},j} = rm_j \cdot \left(1 + \frac{1}{lru_j + \delta}\right), \quad (4)$$

where δ is a very small positive scalar (default 1×10^{-5}). When the model's clustering confidence for the current mention m_i is insufficient (i.e., $\max_j \mathbf{p}_c^j(m_i) \leq 0.5$) and the number of tracked clusters reaches τ_1, the cluster with the smallest $SAES_{\text{train},j}$ is evicted. Otherwise, a new cluster $c_{w+1} = [m_i]$ is created. This design prioritizes clusters with more remaining mentions (i.e., dominated by rm_j), which helps maintain long-distance coreference links. The second term acts as a tie-breaker: when rm_j is tied, a larger lru_j results in a lower score, enforcing LRU eviction. When $lru_j = 0$ (i.e., the cluster was just accessed), the second term is maximized, giving that cluster the highest protection from eviction.

During inference, the true rm_j value is unavailable. Relying solely on LRU may mistakenly remove important clusters. Therefore, we introduce a composite score that combines the number of existing mentions em_j in tracked cluster c_j, the total number of steps since its creation age_j, and lru_j:

$$SAES_{\text{inf},j} = \frac{em_j}{age_j} \cdot \left(1 + \frac{1}{lru_j + \delta}\right). \quad (5)$$

Here, the ratio $\frac{em_j}{age_j}$ reflects the tracked cluster's activity level—a higher value suggests greater coreference potential. Similarly, clusters are sorted by $SAES_{\text{inf},j}$ in ascending order. When clustering confidence is low and the number of tracked clusters reaches τ_1, the cluster with the smallest score is evicted. Moreover, the second term ensures that among clusters with identical mention activity level, the one with the largest lru_j (least recently used) is evicted.

<u>Comment 1.</u> Unlike prior methods that rely solely on LRU (Xia et al., 2020; Toshniwal et al., 2020) or complex dual-cache mechanisms (Guo et al., 2023), SAES employs a unified yet statistics-aware scoring function. It utilizes richer information (global mention counts in training, activity rates in inference) while maintaining a simple, single-cache architecture, thus simplifying management.

3.3 Internal Regularization Policy

To select representative mentions under constrained memory, we introduce an Internal Regularization Policy (**IRP**) that operates on each tracked coreference cluster. This policy limits the number of mentions stored per cluster to a predefined positive integer threshold τ_2. Clusters with at most τ_2 mentions are left unchanged; larger clusters are condensed to their τ_2 most representative mentions. The key challenge is to select a subset of mentions that best preserves the cluster's semantic information. To achieve this, we craft a two-stage algorithm: (1) a grouping strategy based on mention textual relations, and (2) a group-based sampling for selecting representative mentions.

Specifically, ***in the first stage***, we use a Union-Find algorithm to efficiently cluster semantically similar mentions based on textual equivalence/inclusion relationships and a pronoun lexicon (see Appendix B), forming groups that serve as the sampling units. ***In the second stage***, we adopt a group-based sampling strategy. It deterministically retains the first and last mentions of the original cluster to preserve contextual boundaries. For the remaining $\tau_2 - 2$ slots, quotas are dynamically allocated across groups: if the number of groups exceeds available slots, the largest groups are prioritized; otherwise, at least one mention is selected from each group. This strategy maximizes the coverage of diverse semantic units within the condensed cluster. Pseudocode is provided in Appendix A.

<u>Comment 2.</u> The two-stage design of IRP effectively selects a compact yet representative set of mentions from the original cluster. Compared to the existing Transformer-based incremental model (Martinelli et al., 2024), ***IRP is, to our knowledge, a novel design that effectively maintains CR performance under strict memory constraints through its structured information preservation policy.***

<u>Comment 3.</u> Together, IRP and SAES ensure that the representation $\mathbf{M}$ is bounded in size by $\mathbb{R}^{\tau_1 \times (\tau_2+1) \times d_h}$, guaranteeing that the incremental clustering process runs efficiently within the predefined memory budget.

4 Experiments

4.1 Experimental Settings

Datasets. We train and evaluate the performance of our proposed approach on two widely used CR datasets: OntoNotes (Pradhan et al., 2012) and LitBank (Bamman et al., 2020). To assess the generalization capability of approaches, models trained on OntoNotes are also tested on the out-of-domain dataset WikiCoref (Ghaddar and Langlais, 2016). Statistics are detailed in Table 1.

Datasets	#Train	#Val	#Test	Avg. W	Avg. M
OntoNotes	2802	343	348	467	56
LitBank	80	10	10	2105	291
WikiCoref	-	-	30	1996	230

Table 1: Dataset statistics: number of documents in each dataset split (#Train/Val/Test), average number of words (i.e., Avg. W) and mentions (i.e., Avg. M) per document.

Baselines. As outlined in Related Works (see Appendix 5), while numerous CR approaches exist, our work specifically focuses on memory efficiency for incremental clustering. Accordingly, we select the most relevant baselines in the main paper: LRU (Xia et al., 2020; Toshniwal et al., 2020) and Dual-cache (Guo et al., 2023). For more comparison results, please see Appendix D. To gauge the performance of MEIC-DT (which integrates SAES and IRP), we follow Luo et al. (2025b) in adopting Imcoref as Mention Extractor and DeBERTa$_{\text{large}}$ (He et al., 2021) as Text Encoder (see Fig. 2). Furthermore, we conduct coreference clustering experiments adopting two distinct classifiers—a linear classifier (Xia et al., 2020; Toshniwal et al., 2020; Guo et al., 2023) and a lightweight Transformer classifier (Martinelli et al., 2024)—to thoroughly evaluate the effectiveness of MEIC-DT.

Metrics. In our experiments, we evaluate performance using the F1 scores of the MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and CEAF$_{\phi_4}$ (Luo, 2005) metrics. The overall CoNLL-F1 score (abbreviated as Avg.F1) is the average of these three F1 scores. To ensure reliability, we report the average for each experiment over 3 different random seeds.

Configurations. Unless noted otherwise, we set $\tau_1 = 50$ and $\tau_2 = 30$ for the Transformer classifier, and $\tau_1 = 50$ for the linear classifier. For the linear classifier, we follow prior works Xia et al. (2020); Toshniwal et al. (2020); Guo et al. (2023) in representing each historical cluster with a single vector, which eliminates the need for additional parameters beyond τ_1. We set $\delta = 1 \times 10^{-5}$ and employ the Adafactor optimizer (Shazeer and Stern, 2018) for model training, with learning rates configured as 2e-5 for DeBERTa$_{\text{large}}$ and 3e-4 for the remaining model layers. All experiments are implemented using the PyTorch-Lightning framework. Each run is executed on a single RTX 4090 GPU with 24GB of VRAM. Due to space limitations, the complete experimental settings are provided in Appendix C.

4.2 Main Results

This section presents a systematic evaluation and comparative analysis of LRU, Dual-cache, and MEIC-DT across multiple datasets. Note that LRU and Dual-cache only constrain the number of historical coreference clusters (τ_1), whereas MEIC-DT jointly constrains both the number of clusters (τ_1) and the number of mentions per cluster (τ_2) through the synergistic operation of its SAES and IRP modules. To comprehensively assess the performance of these methods, experiments are designed in two phases. First, incremental clustering is performed for $\tau_1 \in \{10, 30, 50, 100, 200\}$ by a linear classifier. Subsequently, under a Transformer architecture, experiments are conducted with $\tau_1/\tau_2 \in \{10/10, 10/20, 10/30, 30/30, 50/30\}$. In particular, since the IRP module is first introduced in this paper, we retain it while replacing the SAES strategy in MEIC-DT with either LRU or Dual-cache, thereby constructing comparable hybrid frameworks. Thus, in the experiments reported here, "LRU+IRP" and "Dual-cache+IRP" actually correspond to LRU and Dual-cache. The overall performance comparison of all methods is presented in Table 2.

LitBank Experiments on the LitBank dataset clearly reveal the combined impact of classifier architecture and constraint strategies. First, the Transformer Classifier consistently and significantly outperforms the Linear Classifier across all approaches. For instance, MEIC-DT achieves an Avg.F1 of 81.27% under $\tau_1/\tau_2 = 50/30$, a substantial improvement over its linear counterpart (77.04%), demonstrating stronger contextual modeling capability. Notably, when using the Linear Classifier, all approaches peak in performance at $\tau_1 = 50$ (with τ_2 as -), while further relaxing constraints to $\tau_1 = 100$ or 200 leads to a drop in Avg.F1. This highlights that classifiers with weaker contextual modeling struggle to effectively utilize

Methods	τ_1/τ_2	Litbank				OntoNotes				WikiCoref			
		MUC	B^3	CEAF$_{\phi_4}$	Avg.F1	MUC	B^3	CEAF$_{\phi_4}$	Avg.F1	MUC	B^3	CEAF$_{\phi_4}$	Avg.F1
Linear Classifier													
Unbound	-/-	84.52$_{\pm0.77}$	65.67$_{\pm0.87}$	76.59$_{\pm0.37}$	75.59$_{\pm0.86}$	89.71$_{\pm0.28}$	77.31$_{\pm0.68}$	69.88$_{\pm0.72}$	78.97$_{\pm0.58}$	72.19$_{\pm0.38}$	53.34$_{\pm0.20}$	64.28$_{\pm0.59}$	63.27$_{\pm0.12}$
LRU	10/-	83.77$_{\pm0.37}$	66.76$_{\pm0.40}$	74.94$_{\pm0.27}$	75.16$_{\pm0.27}$	87.94$_{\pm0.17}$	75.15$_{\pm0.43}$	66.15$_{\pm0.74}$	76.41$_{\pm0.91}$	71.43$_{\pm0.09}$	54.43$_{\pm0.19}$	62.62$_{\pm0.73}$	62.83$_{\pm0.99}$
	30/-	83.91$_{\pm0.19}$	72.38$_{\pm0.83}$	71.37$_{\pm0.14}$	75.89$_{\pm0.32}$	88.49$_{\pm0.38}$	76.05$_{\pm0.92}$	68.05$_{\pm1.84}$	77.53$_{\pm1.02}$	73.01$_{\pm0.63}$	53.80$_{\pm0.42}$	64.08$_{\pm0.98}$	63.63$_{\pm0.80}$
	50/-	84.96$_{\pm0.73}$	66.27$_{\pm0.91}$	77.55$_{\pm0.36}$	**76.26**$_{\pm0.17}$	87.68$_{\pm0.34}$	75.48$_{\pm0.75}$	71.28$_{\pm0.44}$	78.15$_{\pm0.47}$	72.41$_{\pm0.34}$	52.80$_{\pm0.43}$	65.03$_{\pm0.50}$	63.42$_{\pm0.63}$
	100/-	84.76$_{\pm1.38}$	65.14$_{\pm0.82}$	77.35$_{\pm1.20}$	75.75$_{\pm0.97}$	88.76$_{\pm0.31}$	75.85$_{\pm0.31}$	71.14$_{\pm0.17}$	78.58$_{\pm0.13}$	72.35$_{\pm0.10}$	53.95$_{\pm0.01}$	64.23$_{\pm0.92}$	63.51$_{\pm0.83}$
	200/-	84.52$_{\pm0.77}$	65.67$_{\pm0.87}$	76.59$_{\pm1.37}$	75.59$_{\pm0.86}$	8.68$_{\pm0.37}$	75.78$_{\pm0.75}$	71.48$_{\pm0.55}$	**78.65**$_{\pm0.36}$	72.65$_{\pm0.82}$	53.94$_{\pm0.67}$	65.21$_{\pm0.75}$	**63.93**$_{\pm0.43}$
Dual-cache	10/-	84.06$_{\pm1.40}$	66.63$_{\pm0.72}$	73.71$_{\pm0.46}$	74.80$_{\pm0.19}$	87.23$_{\pm0.81}$	75.18$_{\pm1.00}$	70.50$_{\pm0.46}$	77.64$_{\pm0.75}$	71.68$_{\pm0.50}$	54.26$_{\pm0.85}$	61.34$_{\pm0.99}$	62.43$_{\pm0.18}$
	30/-	85.26$_{\pm0.34}$	66.63$_{\pm0.72}$	76.73$_{\pm0.81}$	76.21$_{\pm0.08}$	87.19$_{\pm0.69}$	75.26$_{\pm0.94}$	71.42$_{\pm0.60}$	77.96$_{\pm0.65}$	72.86$_{\pm0.13}$	54.25$_{\pm0.04}$	64.37$_{\pm0.87}$	63.83$_{\pm0.05}$
	50/-	85.42$_{\pm0.55}$	66.46$_{\pm0.98}$	78.20$_{\pm0.27}$	**76.69**$_{\pm0.52}$	88.54$_{\pm0.42}$	76.39$_{\pm0.38}$	71.63$_{\pm0.75}$	**78.85**$_{\pm0.56}$	73.04$_{\pm0.68}$	54.1$_{\pm0.26}$	65.82$_{\pm0.76}$	**64.32**$_{\pm0.90}$
	100/-	84.22$_{\pm0.43}$	65.36$_{\pm0.98}$	77.01$_{\pm0.27}$	75.53$_{\pm0.52}$	88.21$_{\pm0.32}$	75.41$_{\pm0.31}$	71.31$_{\pm0.42}$	78.31$_{\pm0.31}$	71.82$_{\pm0.72}$	53.01$_{\pm0.39}$	64.65$_{\pm0.47}$	63.16$_{\pm0.16}$
	200/-	84.52$_{\pm0.77}$	65.67$_{\pm0.87}$	76.59$_{\pm0.37}$	75.59$_{\pm0.86}$	88.27$_{\pm0.44}$	75.86$_{\pm0.39}$	71.81$_{\pm0.76}$	78.65$_{\pm0.53}$	73.45$_{\pm0.76}$	52.98$_{\pm0.89}$	64.79$_{\pm0.65}$	63.74$_{\pm0.75}$
MEIC-DT	10/-	84.71$_{\pm0.10}$	65.84$_{\pm0.15}$	76.83$_{\pm0.19}$	75.79$_{\pm0.75}$	87.17$_{\pm0.39}$	75.49$_{\pm0.73}$	70.76$_{\pm0.21}$	77.81$_{\pm0.63}$	71.31$_{\pm0.19}$	52.69$_{\pm0.75}$	64.07$_{\pm0.73}$	62.69$_{\pm0.30}$
	Δ	+0.79	-0.85	2.51	+0.81	-0.41	+0.32	+2.44	+0.79	-0.25	-1.66	+2.09	+0.06
	30/-	84.98$_{\pm0.54}$	67.49$_{\pm0.59}$	77.20$_{\pm1.00}$	76.56$_{\pm0.96}$	87.13$_{\pm0.48}$	76.06$_{\pm0.95}$	71.17$_{\pm0.66}$	78.12$_{\pm0.64}$	72.44$_{\pm0.46}$	54.97$_{\pm0.42}$	64.65$_{\pm0.49}$	64.02$_{\pm0.67}$
	Δ	-0.33	+1.12	+0.63	+0.48	-0.71	+0.41	+1.44	+0.38	-0.50	+0.95	+0.43	+0.29
	50/-	85.41$_{\pm0.44}$	67.57$_{\pm0.14}$	78.15$_{\pm0.65}$	**77.04**$_{\pm0.28}$	87.53$_{\pm0.37}$	77.58$_{\pm0.42}$	72.85$_{\pm0.88}$	**79.32**$_{\pm0.67}$	72.84$_{\pm0.43}$	55.06$_{\pm0.42}$	65.64$_{\pm0.47}$	**64.51**$_{\pm0.49}$
	Δ	+0.22	+1.21	+0.28	+0.57	-0.58	+1.65	+1.40	+0.82	+0.12	+1.61	+0.22	+0.64
	100/-	85.56$_{\pm0.05}$	67.46$_{\pm0.11}$	78.02$_{\pm0.42}$	77.01$_{\pm0.46}$	87.95$_{\pm0.42}$	76.17$_{\pm0.41}$	72.96$_{\pm0.85}$	79.03$_{\pm0.59}$	73.02$_{\pm0.19}$	54.94$_{\pm0.27}$	65.50$_{\pm0.22}$	64.49$_{\pm0.41}$
	Δ	+1.07	+2.21	+0.84	+1.37	-0.53	+0.54	+1.73	+0.59	+0.94	+1.46	+1.06	+1.16
	200/-	84.52$_{\pm0.77}$	65.67$_{\pm0.87}$	76.59$_{\pm0.37}$	75.59$_{\pm0.86}$	87.38$_{\pm0.27}$	76.37$_{\pm0.54}$	72.41$_{\pm0.69}$	78.72$_{\pm0.43}$	73.53$_{\pm0.72}$	54.65$_{\pm0.62}$	65.32$_{\pm0.71}$	64.50$_{\pm0.68}$
	Δ	+0.00	+0.00	+0.00	+0.00	-1.10	+0.55	+0.76	+0.07	+0.48	+1.19	+0.32	+0.66
Transformer Classifier													
LRU+IRP	10/10	84.54$_{\pm0.39}$	70.29$_{\pm0.14}$	72.25$_{\pm0.64}$	75.69$_{\pm0.63}$	87.82$_{\pm0.43}$	80.50$_{\pm0.37}$	73.09$_{\pm0.12}$	80.47$_{\pm0.62}$	77.63$_{\pm0.68}$	63.38$_{\pm0.61}$	65.34$_{\pm0.38}$	68.78$_{\pm0.89}$
	10/20	85.36$_{\pm0.76}$	66.11$_{\pm0.94}$	76.40$_{\pm1.24}$	75.96$_{\pm0.97}$	87.75$_{\pm0.01}$	80.80$_{\pm0.26}$	74.82$_{\pm0.69}$	81.12$_{\pm0.33}$	76.97$_{\pm0.10}$	65.44$_{\pm0.26}$	64.47$_{\pm0.39}$	68.96$_{\pm0.08}$
	10/30	84.70$_{\pm0.24}$	73.69$_{\pm0.42}$	71.90$_{\pm0.92}$	76.76$_{\pm0.49}$	88.25$_{\pm0.17}$	81.10$_{\pm0.92}$	78.74$_{\pm0.21}$	82.70$_{\pm0.64}$	77.79$_{\pm0.79}$	66.74$_{\pm0.41}$	64.96$_{\pm0.79}$	69.83$_{\pm0.66}$
	30/30	86.39$_{\pm0.21}$	74.44$_{\pm0.63}$	75.22$_{\pm0.80}$	78.68$_{\pm0.59}$	89.15$_{\pm0.15}$	82.02$_{\pm0.18}$	79.04$_{\pm0.76}$	83.40$_{\pm0.22}$	78.89$_{\pm0.31}$	64.56$_{\pm0.48}$	67.93$_{\pm0.36}$	70.46$_{\pm0.73}$
	50/30	86.68$_{\pm0.71}$	74.96$_{\pm0.13}$	75.95$_{\pm0.64}$	**79.19**$_{\pm0.21}$	89.24$_{\pm0.05}$	82.57$_{\pm0.49}$	79.43$_{\pm0.83}$	**83.75**$_{\pm0.36}$	78.92$_{\pm0.35}$	65.65$_{\pm0.68}$	67.93$_{\pm0.49}$	**70.83**$_{\pm0.65}$
Dual-cache+IRP	10/10	85.40$_{\pm0.18}$	71.81$_{\pm0.21}$	73.52$_{\pm0.64}$	76.91$_{\pm0.89}$	88.41$_{\pm0.34}$	81.11$_{\pm0.78}$	73.25$_{\pm0.69}$	80.92$_{\pm0.59}$	78.46$_{\pm0.29}$	64.88$_{\pm0.46}$	66.57$_{\pm0.42}$	69.97$_{\pm0.35}$
	10/20	85.37$_{\pm0.54}$	72.59$_{\pm0.28}$	74.43$_{\pm0.93}$	77.46$_{\pm0.12}$	88.84$_{\pm0.48}$	81.74$_{\pm0.37}$	75.96$_{\pm0.65}$	82.18$_{\pm0.44}$	78.37$_{\pm0.15}$	65.61$_{\pm0.24}$	67.45$_{\pm0.43}$	70.48$_{\pm0.18}$
	10/30	86.20$_{\pm0.94}$	74.37$_{\pm0.84}$	73.90$_{\pm0.43}$	78.16$_{\pm0.38}$	89.19$_{\pm0.41}$	83.52$_{\pm0.38}$	76.56$_{\pm0.63}$	83.09$_{\pm0.37}$	79.21$_{\pm0.31}$	64.89$_{\pm0.09}$	68.21$_{\pm0.63}$	70.74$_{\pm0.11}$
	30/30	86.89$_{\pm0.10}$	74.79$_{\pm0.11}$	76.03$_{\pm0.59}$	79.24$_{\pm0.15}$	90.18$_{\pm0.10}$	84.27$_{\pm0.44}$	77.35$_{\pm0.56}$	83.93$_{\pm0.27}$	79.17$_{\pm0.09}$	66.67$_{\pm0.46}$	67.83$_{\pm0.61}$	71.22$_{\pm0.32}$
	50/30	87.40$_{\pm0.39}$	77.04$_{\pm0.16}$	77.40$_{\pm0.73}$	**80.61**$_{\pm0.71}$	90.29$_{\pm0.46}$	83.30$_{\pm0.53}$	80.72$_{\pm0.97}$	**84.77**$_{\pm0.64}$	79.73$_{\pm0.24}$	64.30$_{\pm0.20}$	69.83$_{\pm0.41}$	**71.29**$_{\pm0.27}$
MEIC-DT	10/10	86.09$_{\pm0.94}$	72.66$_{\pm0.18}$	75.05$_{\pm0.20}$	77.93$_{\pm0.76}$	88.59$_{\pm0.99}$	81.58$_{\pm0.14}$	74.07$_{\pm0.89}$	81.42$_{\pm0.89}$	79.13$_{\pm0.24}$	65.69$_{\pm0.23}$	68.08$_{\pm0.41}$	70.97$_{\pm0.26}$
	Δ	+1.12	+1.61	+2.17	+1.63	+0.48	+0.77	+0.90	+0.73	+1.09	+1.56	+2.13	+1.60
	10/20	86.46$_{\pm0.94}$	72.63$_{\pm0.16}$	75.52$_{\pm0.40}$	78.20$_{\pm0.16}$	89.23$_{\pm0.58}$	81.75$_{\pm0.19}$	79.48$_{\pm1.00}$	83.49$_{\pm0.46}$	79.33$_{\pm0.24}$	65.50$_{\pm0.53}$	68.40$_{\pm0.39}$	71.08$_{\pm0.26}$
	Δ	+1.82	+0.15	+2.62	+1.52	+0.94	+0.48	+4.09	+1.84	+1.66	-0.02	+2.44	+1.36
	10/30	86.55$_{\pm0.33}$	73.92$_{\pm0.91}$	75.37$_{\pm0.65}$	78.61$_{\pm0.92}$	89.40$_{\pm0.89}$	82.40$_{\pm0.40}$	79.58$_{\pm0.39}$	83.80$_{\pm0.56}$	79.41$_{\pm0.38}$	66.73$_{\pm0.45}$	68.21$_{\pm0.32}$	71.45$_{\pm0.31}$
	Δ	+1.10	-0.11	+2.47	+1.15	+0.68	+0.09	+1.93	+0.90	+0.91	+0.92	+1.63	+1.17
	30/30	86.86$_{\pm0.18}$	74.77$_{\pm0.99}$	75.91$_{\pm0.79}$	79.18$_{\pm0.08}$	90.05$_{\pm0.46}$	84.49$_{\pm0.31}$	75.24$_{\pm0.16}$	84.26$_{\pm0.63}$	79.73$_{\pm0.18}$	67.69$_{\pm0.33}$	68.90$_{\pm0.60}$	72.11$_{\pm0.38}$
	Δ	+0.22	+0.16	+0.28	+0.22	+0.38	+1.35	+0.04	+0.59	+0.70	+2.08	+1.02	+1.27
	50/30	87.85$_{\pm0.28}$	78.53$_{\pm0.24}$	77.42$_{\pm0.03}$	**81.27**$_{\pm0.41}$	90.53$_{\pm0.43}$	84.43$_{\pm0.42}$	80.19$_{\pm0.49}$	**85.05**$_{\pm0.67}$	80.16$_{\pm0.25}$	68.38$_{\pm0.21}$	69.35$_{\pm0.43}$	**72.63**$_{\pm0.24}$
	Δ	+0.31	+2.53	-0.26	+0.87	+0.77	+0.50	+1.11	+0.79	+0.83	+3.40	+0.47	+1.57

Table 2: Results of different approaches on varying datasets. We report the MUC, B^3, and CEAF$_{\phi_4}$ F1 scores (%), their average (Avg. F1). Of note, **Bold** and underline highlight the best Avg.F1 within each approach (with varying τ_1/τ_2) and the best overall Avg.F1, respectively. Δ measures the average gain of our method over baselines: $\Delta = \frac{(a-b)+(a-c)}{2}$ (a: our method, b, c: baselines). "Unbound" means no constraint on cluster count for the linear classifier.

excessively large memory capacity. Second, under different τ_1/τ_2 settings, the performance of all approaches follows a regular pattern: more relaxed constraints (e.g., $\tau_1 = 50, \tau_2 = 30$) generally yield the best results, indicating that providing moderate memory capacity is beneficial for this dataset. Finally, compared to baseline approaches, MEIC-DT significantly outperforms LRU+IRP (79.19%) and Dual-cache+IRP (80.61%) under identical constraint settings (e.g., $50/30$), with its Δ gains consistently positive. This validates the advantage of the synergistic optimization between SAES and IRP, particularly evident in key metrics such as B^3.

OntoNotes Results on OntoNotes further consolidate the effectiveness of our approach. The Transformer Classifier again demonstrates superior performance over the Linear Classifier, with MEIC-DT reaching a peak Avg.F1 of 85.05% under the Transformer, higher than the 79.32% of its linear version. Similarly, under the Linear Classifier, most approaches achieve their optimal Avg.F1 at $\tau_1 = 50$ (except LRU), with performance declining under more relaxed settings, further confirming the necessity of appropriate constraints. Regarding the tuning of τ_1/τ_2, an interesting observation is that under tighter memory constraints (e.g., $10/10$), the improvement of MEIC-DT over baselines (the Δ value) is particularly significant, indicating that its dual-threshold mechanism better maintains performance stability in resource-constrained scenarios. In direct comparison with existing baselines, MEIC-DT leads LRU+IRP and Dual-cache+IRP in Avg.F1 across almost all tested τ_1/τ_2 combinations, showing robust performance especially on the CEAF$_{\phi_4}$ metric, which highlights its improvements in clustering quality and consistency.

WikiCoref On the cross-domain generalization test set WikiCoref, this experiment evaluates the adaptability of models trained on OntoNotes. The results again confirm the effective generalization of the Transformer Classifier, with its performance across all approaches far surpassing that of the corresponding Linear Classifier versions. Even in out-of-domain testing, the Linear Classifier exhibits a

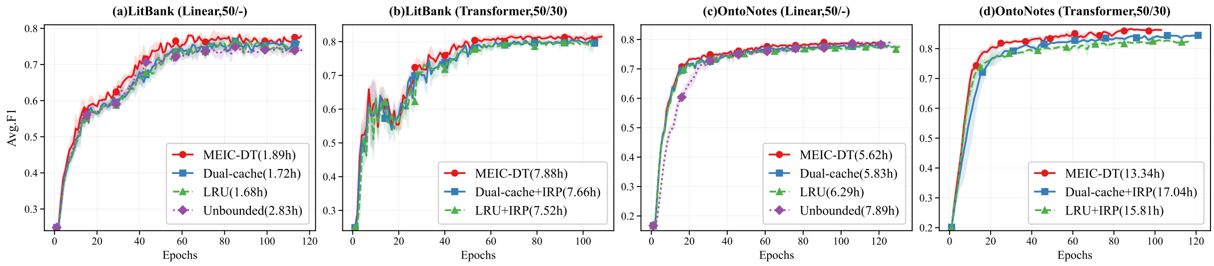

Figure 3: Learning curves and total training time. Results are shown for configurations with $\tau_1/\tau_2 = 50/-$ and $50/30$, across different datasets (LitBank, OntoNotes) and classifier backbones.

similar trend, performing best at $\tau_1 = 50$ before degrading, underscoring the stability of the Transformer architecture in complex scenarios. Regarding constraint parameters, MEIC-DT achieves an optimal Avg.F1 of 72.63% under $\tau_1/\tau_2 = 50/30$, suggesting that a balanced constraint configuration helps the model capture cross-domain coreference structures. More importantly, compared to the two baselines, MEIC-DT achieves the most significant lead on this out-of-domain test set (e.g., outperforming the best baseline, Dual-cache+IRP, by approximately 1.34 Avg.F1 points). Its Δ values are positive and substantial in most settings, strongly demonstrating that the mechanisms proposed in MEIC-DT possess superior domain generalization capability and robustness.

τ_1/τ_2	Litbank TAM	Litbank IAM	Litbank IAT	OntoNotes TAM	OntoNotes IAM	OntoNotes IAT	WikiCoref IAM	WikiCoref IAT
Linear Classifier								
-/-	255.44	19.87	0.46	150.31	19.83	0.31	19.86	0.42
10/-	72.22	19.15	0.62	52.86	19.14	0.37	19.15	0.46
30/-	153.31	19.34	0.64	89.43	19.34	0.43	19.35	0.49
50/-	207.14	19.53	0.69	99.23	19.51	0.47	19.52	0.53
100/-	251.24	19.83	0.72	120.54	19.78	0.53	19.81	0.58
200/-	255.44	19.87	0.46	147.37	19.81	0.55	19.84	0.59
Transformer Classifier								
10/10	975.93	20.16	1.12	616.58	19.59	0.67	19.82	0.89
10/20	1628.74	21.83	1.15	964.71	20.98	0.71	21.62	1.01
10/30	2118.95	23.43	1.22	1166.23	22.46	0.73	22.89	1.13
30/30	7045.07	36.55	1.41	3046.73	31.93	0.77	36.31	1.29
50/30	14581.14	38.68	1.81	7635.27	38.33	0.80	38.49	1.68

Table 3: Average resource consumption under identical settings. Note: Metrics are averaged across different approaches for each fixed configuration of dataset, classifier, and τ_1/τ_2. TAM, IAM, and IAT denote Training Average Memory (MB), Inference Average Memory (MB), and Inference Average Time (s), respectively.

4.3 Speed and Memory Usage

Our experiments reveal that under identical configurations of dataset, classifier, and τ_1/τ_2 constraints, different approaches exhibit negligible differences in terms of Training Average Memory (TAM), Inference Average Memory (IAM), and Inference Average Time (IAT). Therefore, we report the averaged values of these metrics under each such fixed setting in Table 3. In contrast, the training dynamics and time consumption vary significantly across approaches. Figure 3 presents the learning curves and total training time for various approaches under the τ_1/τ_2 settings of $50/-$ and $50/30$, across different datasets and classifiers.

As shown in Table 3, the Transformer Classifier incurs significantly higher TAM than the Linear Classifier due to its more complex architecture. However, the gap in IAM and IAT between the two is relatively small, indicating that the constraint configuration (τ_1/τ_2) is the dominant factor for inference cost. As τ_1/τ_2 increases, both IAM and IAT rise monotonically across all configurations. Notably, under the Transformer architecture, memory consumption surges sharply from $30/30$ to $50/30$. This trend directly aligns with the performance peak observed at $50/30$ (see Table 2), clearly revealing the inherent trade-off between performance and efficiency. Figure 3 provides a deeper analysis of the training dynamics. On LitBank, MEIC-DT slightly increases training time compared to baselines, yet its learning curve consistently dominates, indicating higher convergence efficiency. On OntoNotes, MEIC-DT demonstrates a more pronounced advantage: it not only requires substantially less training time (e.g., only 13.34 hours with the Transformer backbone, significantly lower than the baselines' 17.04 and 15.81 hours) and fewer epochs but also exhibits a learning curve that is superior throughout or in later stages. Together, these results demonstrate that the dual-threshold mechanism of MEIC-DT not only enhances final performance (see Table 2) but also accelerates model convergence through a more efficient and stable optimization process, achieving a dual advantage in both effectiveness and efficiency.

4.4 Ablation Study

The Efficacy of SAES. We look into the efficacy of SAES in Litbank and OntoNotes using Transformer classifier with $\tau_1/\tau_2 = 50/30$. As detailed in Section 3.2, MEIC-DT employs distinct eviction criteria for training and inference, termed $\text{SAES}_{\text{train}}$ and SAES_{inf}. To ablate the contribution of each stage, we construct hybrid variants by replacing the corresponding SAES module with either LRU or Dual-cache during training or inference. We report results in Table 4. As shown in Table 4, employing the $\text{SAES}_{\text{train}}/\text{SAES}_{\text{inf}}$ strategy yields the highest Avg.F1 on both LitBank and OntoNotes. Replacing SAES_{inf} with LRU during inference alone causes a marked performance drop (e.g., from 81.27% to 79.43% on LitBank), underscoring the effectiveness of SAES_{inf} in preserving critical clusters via mention activity (em_j/age_j) in the absence of gold annotations. Similarly, substituting $\text{SAES}_{\text{train}}$ with LRU or Dual-cache in training also degrades results, validating the superiority of its eviction criterion based on gold remaining mentions (rm_j). These results collectively demonstrate that the stage-specific strategies of SAES—tailored to the distinct information available during training and inference—are both indispensable, and their synergy is crucial to MEIC-DT's leading performance.

Train/Inference	MUC	B^3	CEAF_{ϕ_4}	Avg.F1
Litbank				
$\text{SAES}_{\text{train}}/\text{SAES}_{\text{inf}}$	87.85$_{\pm 0.28}$	78.53$_{\pm 0.24}$	77.42$_{\pm 0.03}$	**81.27**$_{\pm 0.41}$
$\text{SAES}_{\text{train}}$/LRU	86.87$_{\pm 0.54}$	75.42$_{\pm 0.31}$	76.01$_{\pm 0.22}$	79.43$_{\pm 0.25}$
$\text{SAES}_{\text{train}}$/Dual-cache	87.56$_{\pm 0.46}$	77.08$_{\pm 0.28}$	77.34$_{\pm 0.62}$	80.66$_{\pm 0.39}$
LRU/SAES_{inf}	86.17$_{\pm 0.51}$	75.63$_{\pm 0.11}$	75.06$_{\pm 0.38}$	79.15$_{\pm 0.21}$
Dual-cache/SAES_{inf}	87.02$_{\pm 0.38}$	76.99$_{\pm 0.26}$	76.89$_{\pm 0.14}$	80.30$_{\pm 0.27}$
$\text{SAES}_{\text{inf}}/\text{SAES}_{\text{inf}}$	87.45$_{\pm 0.42}$	77.29$_{\pm 0.48}$	77.38$_{\pm 0.38}$	80.71$_{\pm 0.67}$
OntoNotes				
$\text{SAES}_{\text{train}}/\text{SAES}_{\text{inf}}$	90.53$_{\pm 0.43}$	84.43$_{\pm 0.42}$	80.19$_{\pm 0.49}$	**85.05**$_{\pm 0.67}$
$\text{SAES}_{\text{train}}$/LRU	89.29$_{\pm 0.54}$	82.57$_{\pm 0.14}$	79.25$_{\pm 0.35}$	83.71$_{\pm 0.16}$
$\text{SAES}_{\text{train}}$/Dual-cache	90.31$_{\pm 0.32}$	83.46$_{\pm 0.24}$	80.69$_{\pm 0.14}$	84.82$_{\pm 0.52}$
LRU/SAES_{inf}	89.48$_{\pm 0.17}$	82.52$_{\pm 0.26}$	79.41$_{\pm 0.33}$	83.80$_{\pm 0.29}$
Dual-cache/SAES_{inf}	90.33$_{\pm 0.58}$	82.35$_{\pm 0.47}$	80.89$_{\pm 0.34}$	84.86$_{\pm 0.71}$
$\text{SAES}_{\text{inf}}/\text{SAES}_{\text{inf}}$	90.28$_{\pm 0.61}$	83.45$_{\pm 0.57}$	79.96$_{\pm 0.38}$	84.89$_{\pm 0.55}$

Table 4: The Impact (%) of the SAES Module in MEIC-DT. Evaluation on LitBank and OntoNotes with the Transformer classifier under $\tau_1/\tau_2 = 50/30$.

IRP	MUC	B^3	CEAF_{ϕ_4}	Avg.F1
Litbank				
Group-based Sampling (ours)	87.85$_{\pm 0.28}$	78.53$_{\pm 0.24}$	77.42$_{\pm 0.03}$	**81.27**$_{\pm 0.41}$
Random Sampling	87.12$_{\pm 0.15}$	76.98$_{\pm 0.27}$	77.01$_{\pm 0.13}$	80.37$_{\pm 0.22}$
OntoNotes				
Group-based Sampling (ours)	90.53$_{\pm 0.43}$	84.43$_{\pm 0.42}$	80.19$_{\pm 0.49}$	**85.05**$_{\pm 0.67}$
Random Sampling	89.08$_{\pm 0.22}$	82.33$_{\pm 0.54}$	80.04$_{\pm 0.19}$	83.82$_{\pm 0.37}$

Table 5: The utility (%) of the IRP Module in MEIC-DT. Evaluation on LitBank and OntoNotes with the Transformer classifier under $\tau_1/\tau_2 = 50/30$.

The utility of IRP. We conduct an ablation study to assess the utility of the IRP module in MEIC-DT on OntoNotes and LitBank, using the Transformer classifier under $\tau_1/\tau_2 = 50/30$. As introduced in Section 3.3, IRP is a core innovation that employs ***group-based sampling*** by leveraging mention textual relations to select the most representative mentions per cluster. To validate this strategy, we compare it against a ***random sampling*** baseline, which selects mentions uniformly at random from each cluster without grouping. Results are presented in Table 5. From 5, the group-based sampling significantly outperforms random sampling on both LitBank and OntoNotes (e.g., achieving a +1.23 gain in Avg.F1 on OntoNotes). This confirms that IRP can more effectively preserve the core semantic information of each cluster through semantic-relation-based grouping and stratified sampling. The semantic space visualization in Fig. 4 further reveals that the mentions selected by group-based sampling exhibit a more reasonable and representative distribution, which directly explains its superiority in maintaining CR performance and validates the key role of IRP as a structured information compression strategy.

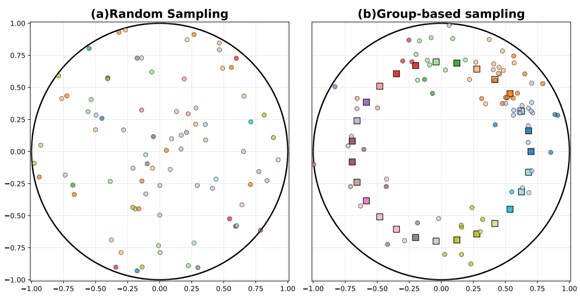

Figure 4: An example of semantic space distributions for different sampling strategies on OntoNotes.

5 Related Works

Coreference Resolution (CR) has long been a crucial task in the field of NLP. The seminal work by (Lee et al., 2017) introduced the first end-to-end supervised neural method, i.e., the Coarse-to-Fine model, establishing the detect-then-cluster pipeline. Shortly thereafter, a panoply of efforts have focused on improving this method's training efficiency, memory consumption, and coreference performance. For example, some methods (Dobrovolskii, 2021; Si et al., 2022; Kantor and Globerson, 2019; Xu and Choi, 2020; Wu et al., 2020; Kirstain et al., 2021; Lai et al., 2022; Otmazgin et al., 2022a,b; D'Oosterlinck et al., 2023; Sundar et al., 2024) retained the detect-then-cluster pipeline and leveraged pre-trained language models (e.g., BERT (Devlin et al., 2019), SpanBERT (Joshi et al., 2020), Longformer (Beltagy et al., 2020)) for document encoding, significantly boosting performance. However, they still grapple with substantial memory overhead. Also, generative methods (Liu et al., 2022; Bohnet et al., 2023; Zhang et al., 2023), centered around large sequence-to-sequence architectures, once dominated the pursuit of high coreference performance. Nevertheless, their prohibitively expensive training costs and high inference latency render their deployment infeasible.

With super-sized training corpora and computational cluster resources, LLMs have demonstrated powerful reasoning capabilities, thus enabling SOTA performance in a wide range of natural language tasks (Achiam et al., 2023; Liu et al., 2024a; Team, 2025; Luo et al., 2024; Tan et al., 2024; Yu et al., 2024; Si et al., 2025c; Wang et al., 2025; Si et al., 2025b; Luo et al., 2025a; Si et al., 2025a, 2024; Bai et al.). However, their performance in CR has yet to surpass mainstream supervised neural methods (Le and Ritter, 2023; Gan et al., 2024). This limitation stems from the "large but not precise" nature of LLMs, which hinders high-accuracy mention extraction (Liu et al., 2024b). Intriguingly, under ideal conditions where gold mentions are provided, LLMs leverage their strong reasoning capabilities to achieve CR performance that matches or even exceeds that of supervised neural methods (Le and Ritter, 2023). Recently, other attempts to incorporate LLMs into CR either rely on overly idealistic assumptions (Sundar et al., 2024) or entail prohibitive computational costs to train multiple LLMs (Zhu et al., 2025).

Of note, under *supervised neural methods*, there exists another line of works (Xia et al., 2020; Toshniwal et al., 2020; Guo et al., 2023; Martinelli et al., 2024) to drew inspiration from human cognitive incremental processing mechanisms for long-text scenarios. During the mention clustering, they progressively evaluate the association between candidate mentions and existing coreference clusters using linear classifiers or lightweight Transformer architectures to determine coreference links. However, despite significant progress in CR, they still face a critical challenge in balancing efficiency and performance during mention clustering. For example, conventional incremental clustering approaches (Xia et al., 2020; Toshniwal et al., 2020; Guo et al., 2023) typically employ a fixed-size cache and a linear classifier to associate candidate mentions with coreference clusters (each represented as a single vector) tracked by the cache. These approaches decide whether a mention should join an existing cluster or form a new one. Upon reaching capacity, a predefined rule-based eviction strategy—LRU (*Least Recently Used*) or a dual-cache mechanism combining LRU and LFU (*Least Frequently Used*)—removes certain entities to update the cache dynamically. While computationally efficient, such approaches often fail to capture the complex nonlinear dependencies between mentions and clusters, and yield limited semantic representations due to excessive intra-cluster compression.

Moreover, these strategies either rely on a single heuristic (e.g., LRU) that may discard essential clusters, or employ a dual-cache mechanism that increases the complexity of cache management. Crucially, the same eviction strategy is used across both training and inference. This does not account for the inherent differences in information availability between the two phases and thus constrains the model's overall effectiveness. Recently, Martinelli et al. (2024) introduced a lightweight Transformer model. Its input is formed by concatenating the representation of a candidate mention with representations of all mentions from historical coreference clusters, effectively modeling the global intra-cluster structure and significantly improving clustering accuracy. However, a major drawback is that the input size grows polynomially with the number of clusters and the number of mentions per cluster, causing memory usage and computational latency to surge during training, which severely limits its practicality and scalability given academic computational resources.

6 Conclusion

This paper presents MEIC-DT, a memory-efficient incremental clustering model for CR that operates under strict dual-threshold memory constraints. The core of our approach is a dual-threshold constraint mechanism, instantiated through two novel components: a Statistics-Aware Eviction Strategy for phase-adaptive cache management, and an Internal Regularization Policy for semantically representative cluster condensation. Extensive experiments show that MEIC-DT achieves a superior balance between performance and memory efficiency compared to existing baselines, validating the effectiveness of its synergistic design in managing the inherent trade-offs for long-text processing.

Limitations

While our method demonstrates significant advances in memory-efficient incremental clustering for coreference resolution, several limitations warrant discussion. ***First***, in the current era dominated by large language models (LLMs), our work—focusing on supervised neural methods—might appear to swim against the tide. However, we note that recent studies (Liu et al., 2024b) have shown high-performing supervised coreference resolvers can effectively enhance LLMs' capabilities in downstream tasks such as generation and question answering, which presents a clear and valuable application niche for our efficient model. ***Second***, our experiments were conducted under constrained computational budgets. We acknowledge that with more generous resources (e.g., larger τ_1/τ_2 or model capacity), performance could potentially be further improved. ***Finally***, we believe more effective memory constraint strategies may exist. Future work could explore adaptive threshold learning or reinforcement learning-based policies for dynamic memory allocation.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.T2341003), Beijing Municipal Science and Technology Plan Project (Z241100001324025) and a grant from the Guoqiang Institute, Tsinghua University.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Gerry Altmann and Mark Steedman. 1988. Interaction with context during human sentence processing. *Cognition*, 30(3):191–238.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer.

Yuzhuo Bai, Kangyang Luo, Wenhao Li, Shuzheng Si, Gang Chen, Fanchao Qi, and Maosong Sun. Sisfact: Towards systematic, interpretable and scalable factuality evaluation for llm. In *Knowledgeable Foundation Models at ACL 2025*.

David Bamman, Olivia Lewke, and Anya Mansoor. 2020. An annotated dataset of coreference in English literature. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 44–54, Marseille, France. European Language Resources Association.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Bernd Bohnet, Chris Alberti, and Michael Collins. 2023. Coreference resolution through a seq2seq transition-based system. *Transactions of the Association for Computational Linguistics*, 11:212–226.

Ruirui Chen, Weifeng Jiang, Chengwei Qin, Bo Xiong, Fiona Liausvia, Dongkyu Choi, and Boon Kiat Quek. 2025. Are large language models effective knowledge graph constructors? *arXiv preprint arXiv:2510.11297*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Vladimir Dobrovolskii. 2021. Word-level coreference resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7670–7675, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Karel D'Oosterlinck, Semere Kiros Bitew, Brandon Papineau, Christopher Potts, Thomas Demeester, and Chris Develder. 2023. Caw-coref: conjunction-aware word-level coreference resolution. *arXiv preprint arXiv:2310.06165*.

Yujian Gan, Massimo Poesio, and Juntao Yu. 2024. Assessing the capabilities of large language models in coreference: An evaluation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1645–1665, Torino, Italia. ELRA and ICCL.

Abbas Ghaddar and Phillippe Langlais. 2016. WikiCoref: An English coreference-annotated corpus of Wikipedia articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 136–142, Portorož, Slovenia. European Language Resources Association (ELRA).

Qipeng Guo, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2023. Dual cache for long document neural coreference resolution. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15272–15285, Toronto, Canada. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Youngjoon Jang, Seongtae Hong, Junyoung Son, Sungjin Park, Chanjun Park, and Heuiseok Lim. 2025. From ambiguity to accuracy: The transformative effect of coreference resolution on retrieval-augmented generation systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 422–433, Vienna, Austria. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8:64–77.

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677.

Lauri Karttunen. 1976. Discourse referents. In *Notes from the linguistic underground*, pages 363–385. Brill.

Yuval Kirstain, Ori Ram, and Omer Levy. 2021. Coreference resolution without span representations. *arXiv preprint arXiv:2101.00434*.

Tuan Manh Lai, Trung Bui, and Doo Soon Kim. 2022. End-to-end neural coreference resolution revisited: A simple yet effective baseline. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8147–8151. IEEE.

Nghia T Le and Alan Ritter. 2023. Are large language models robust coreference resolvers? *arXiv preprint arXiv:2305.14489*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. *arXiv preprint arXiv:1804.05392*.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, and 1 others. 2024a. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.

Tianyu Liu, Yuchen Eleanor Jiang, Nicholas Monath, Ryan Cotterell, and Mrinmaya Sachan. 2022. Autoregressive structured prediction with language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 993–1005, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yanming Liu, Xinyue Peng, Jiannan Cao, Shi Bo, Yanxin Shen, Tianyu Du, Sheng Cheng, Xun Wang, Jianwei Yin, and Xuhong Zhang. 2024b. Bridging context gaps: Leveraging coreference resolution for long contextual understanding. *arXiv preprint arXiv:2410.01671*.

Kangyang Luo, Yuzhuo Bai, Cheng Gao, Shuzheng Si, Yingli Shen, Zhu Liu, Zhitong Wang, Cunliang Kong, Wenhao Li, Yufei Huang, and 1 others. 2025a. Gltw: Joint improved graph transformer and llm via three-word language for knowledge graph completion. *arXiv preprint arXiv:2502.11471*.

Kangyang Luo, Yuzhuo Bai, Shuzheng Si, Cheng Gao, Zhitong Wang, Yingli Shen, Wenhao Li, Zhu Liu, Yufeng Han, Jiayi Wu, and 1 others. 2025b. Imcorefces: An improved lightweight pipeline for coreference resolution with llm-based checker-splitter refinement. *arXiv preprint arXiv:2510.10241*.

Kangyang Luo, Zichen Ding, Zhenmin Weng, Lingfeng Qiao, Meng Zhao, Xiang Li, Di Yin, and Jinlong Shu. 2024. Let's be self-generated via step by step: A curriculum learning approach to automated reasoning with large language models. *arXiv preprint arXiv:2410.21728*.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 25–32.

Giuliano Martinelli, Edoardo Barba, and Roberto Navigli. 2024. Maverick: Efficient and accurate coreference resolution defying recent trends. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13380–13394, Bangkok, Thailand. Association for Computational Linguistics.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2022a. F-coref: Fast, accurate and easy to use coreference resolution. *arXiv preprint arXiv:2209.04280*.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2022b. Lingmess: Linguistically informed multi expert scorers for coreference resolution. *arXiv preprint arXiv:2205.12644*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on*

EMNLP and CoNLL - Shared Task, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Yu-Ming Shang, Hongli Mao, Tian Tian, Heyan Huang, and Xian-Ling Mao. 2025. From local to global: Leveraging document graph for named entity recognition. *Knowledge-Based Systems*, 312:113017.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Shuzheng Si, Shuang Zeng, Jiaxing Lin, and Baobao Chang. 2022. SCL-RAI: Span-based contrastive learning with retrieval augmented inference for unlabeled entity problem in NER. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2313–2318, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Shuzheng Si, Haozhe Zhao, Gang Chen, Cheng Gao, Yuzhuo Bai, Zhitong Wang, Kaikai An, Kangyang Luo, Chen Qian, Fanchao Qi, and 1 others. 2025a. Aligning large language models to follow instructions and hallucinate less via effective data filtering. *arXiv preprint arXiv:2502.07340*.

Shuzheng Si, Haozhe Zhao, Gang Chen, Yunshui Li, Kangyang Luo, Chuancheng Lv, Kaikai An, Fanchao Qi, Baobao Chang, and Maosong Sun. 2024. Gateau: Selecting influential samples for long context alignment. *arXiv preprint arXiv:2410.15633*.

Shuzheng Si, Haozhe Zhao, Cheng Gao, Yuzhuo Bai, Zhitong Wang, Bofei Gao, Kangyang Luo, Wenhao Li, Yufei Huang, Gang Chen, and 1 others. 2025b. Teaching large language models to maintain contextual faithfulness via synthetic tasks and reinforcement learning. *arXiv preprint arXiv:2505.16483*.

Shuzheng Si, Haozhe Zhao, Kangyang Luo, Gang Chen, Fanchao Qi, Minjia Zhang, Baobao Chang, and Maosong Sun. 2025c. A goal without a plan is just a wish: Efficient and effective global planner training for long-horizon agent tasks. *arXiv preprint arXiv:2510.05608*.

Kawshik Manikantan Sundar, Shubham Toshniwal, Makarand Tapaswi, and Vineet Gandhi. 2024. Major entity identification: A generalizable alternative to coreference resolution. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11679–11695, Miami, Florida, USA. Association for Computational Linguistics.

Keren Tan, Kangyang Luo, Yunshi Lan, Zheng Yuan, and Jinlong Shu. 2024. An llm-enhanced adversarial editing system for lexical simplification. *arXiv preprint arXiv:2402.14704*.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Shubham Toshniwal, Sam Wiseman, Allyson Ettinger, Karen Livescu, and Kevin Gimpel. 2020. Learning to Ignore: Long Document Coreference with Bounded Memory Neural Networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8519–8526, Online. Association for Computational Linguistics.

Shubham Toshniwal, Patrick Xia, Sam Wiseman, Karen Livescu, and Kevin Gimpel. 2021. On generalization in coreference resolution. *arXiv preprint arXiv:2109.09667*.

Marc Vilain, John D Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Zhitong Wang, Cheng Gao, Chaojun Xiao, Yufei Huang, Shuzheng Si, Kangyang Luo, Yuzhuo Bai, Wenhao Li, Tangjian Duan, Chuancheng Lv, and 1 others. 2025. Document segmentation matters for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8063–8075.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963, Online. Association for Computational Linguistics.

Patrick Xia, João Sedoc, and Benjamin Van Durme. 2020. Incremental neural coreference resolution in constant memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8617–8624, Online. Association for Computational Linguistics.

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8527–8533, Online. Association for Computational Linguistics.

Jianxiang Yu, Zichen Ding, Jiaqi Tan, Kangyang Luo, Zhenmin Weng, Chenghua Gong, Long Zeng, Renjing Cui, Chengcheng Han, Qiushi Sun, and 1 others. 2024. Automated peer reviewing in paper sea: Standardization, evaluation, and analysis. *arXiv preprint arXiv:2407.12857*.

Wenzheng Zhang, Sam Wiseman, and Karl Stratos. 2023. Seq2seq is all you need for coreference resolution. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11493–11504, Singapore. Association for Computational Linguistics.

Lixing Zhu, Jun Wang, and Yulan He. 2025. Llm-Link: Dual LLMs for dynamic entity linking on long narratives with collaborative memorisation and prompt optimisation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11334–11347, Abu Dhabi, UAE. Association for Computational Linguistics.

Appendix

This appendix is organized as follows.

- In Section A, we present the detailed algorithmic pseudocode for Internal Regularization Policy.
- In Section B, we list all the pronouns.
- In Section C, we provide a comprehensive description of the experimental setup, including the datasets employed, baseline methods compared, evaluation metrics utilized, and detailed implementation configurations.
- In Section D, we present additional experimental results.

A Pseudocode of Internal Regularization Policy

In this section, we present the detailed algorithmic pseudocode for Internal Regularization Policy, as shown in Algorithms 1-3.

Algorithm 1 Internal Regularization Policy (**IRP**)

Require: Cluster set $\mathcal{C} = \{c_1, c_2, \ldots, c_{\tau_1}\}$ where each $c_i = [(t^1_{i,s}, t^1_{i,e}), (t^2_{i,s}, t^2_{i,e}), \ldots]$ represents mention spans, original word text $words$
Require: Maximum cluster size threshold τ_2
Ensure: Compressed cluster set $\mathcal{C}'$ where each cluster has at most τ_2 mentions

```
for each cluster c_i ∈ C do
    if |c_i| > τ_2 then
        Sort c_i in ascending order, i.e., sort(c_i)
        Initialize compressed cluster: c'_i ← [c_i[0]] {Preserve first mention}
        ĉ_i ← c_i[1 : −1] {Exclude first and last mentions}
        Ĉ_i^grouped ← SG(words, ĉ_i)
        ĉ_i^sampled ← GS(Ĉ_i^grouped, τ_2 − 2)
        c'_i.extend(ĉ_i^sampled)
        c'_i.append(c_i[−1]) {Preserve last mention}
        C' ← C' ∪ {sort(c'_i)}
    else
        C' ← C' ∪ {c_i} {Keep original cluster}
    end if
end for
return C'
```

B Full Pronouns List

We list all the pronouns in Table 6.

i , me , my , mine , myself , you , your , yours , yourself , yourselves , he , him , his , himself , she , her , hers , herself , it , its , itself , we , us , our , ours , ourselves , they , them , their , themselves , that , this

Table 6: Full pronouns list. Note that in the specific experimental process, "it" should be removed from full pronouns list because "it" may lack referential significance.

C Complete Experimental Settings

Datasets and Baselines. We train and evaluate the performance of our proposed methods on two widely used CR datasets: OntoNotes (Pradhan et al., 2012) and LitBank (Bamman et al., 2020). Specifically, OntoNotes, originating from the CoNLL-2012 shared task, serves as the de facto standard for evaluating CR methods. It encompasses seven text genres, including full-length documents such as newswire articles, broadcast news, magazines, web text, and testament passages, as well as multi-speaker transcripts like broadcast conversations and telephone conversations. LitBank is frequently employed for evaluating CR on long documents and comprises 100 literary works. Note that OntoNotes and LitBank follow different annotation guidelines: OntoNotes does not annotate singleton clusters (i.e., single-mention clusters), whereas LitBank provides such annotations. To assess the generalization capability of methods, models trained on OntoNotes are also tested on the out-of-domain dataset WikiCoref (Ghaddar and Langlais, 2016). WikiCoref only contains 30 Wikipedia articles as its test set, with some texts reaching lengths of up to 9869 words. We detail these datasets in Table 7.

Meanwhile, to gauge the effectiveness of MEIC-DT, we select the most relevant baselines in the main paper: LRU (Xia et al., 2020; Toshniwal et al., 2020) and Dual-cache (Guo et al., 2023). We follow Luo et al. (2025b) in adopting Imcoref as Mention Extractor and $\text{DeBERTa}_{\text{large}}$ (He et al., 2021) as Text Encoder (see Fig. 2). Furthermore, we conduct coreference clustering experiments adopting two distinct classifiers—a linear classifier (Xia et al., 2020; Toshniwal et al., 2020; Guo et al., 2023) and a lightweight Transformer classifier (Martinelli et al., 2024)—to thoroughly evaluate the effectiveness of MEIC-DT. As outlined in Related Works (see Appendix 5), numerous CR approaches exist. Accordingly, for more comparison

Algorithm 2 Semantic Grouping (**SG**)

Require: $words$, $\hat{c}$: list of mention spans $[(t_{1,s}, t_{1,e}), (t_{2,s}, t_{2,e}), \ldots, (t_{|\hat{c}|-1,s}, t_{|\hat{c}|-1,e})]$, and $\mathcal{P}$: pronoun lexicon (see Appendix B)
Ensure: $\hat{\mathcal{C}}^{grouped}$: semantically grouped mention spans

$mentions \leftarrow [\,]$ {Store mention texts}
for each $(t_s, t_e) \in \hat{c}$ **do**
    $span = []$
    for $idx \in [t_s, t_e]$ **do**
        $span.append(words[idx].lower())$
    end for
    $mentions.append(\text{concat}(span))$
end for
$n \leftarrow |\hat{c}|$
$uf \leftarrow \text{UnionFind}(n)$ {Initialize Union-Find data structure}
for $i = 0$ to $n - 1$ **do**
    for $j = i$ to $n - 1$ **do**
        if $mentions[i] \in \mathcal{P} \wedge mentions[j] \in \mathcal{P}$ **then**
            if $mentions[i] = mentions[j]$ **then**
                $uf.union(i, j)$
            end if
        else if $mentions[i] \notin \mathcal{P} \wedge mentions[j] \notin \mathcal{P}$ **then**
            if $mentions[i] = mentions[j] \vee mentions[i] \subseteq mentions[j] \vee mentions[j] \subseteq mentions[i]$ **then**
                $uf.union(i, j)$
            end if
        end if
    end for
end for
Initialize $groups \leftarrow \{\}$
for $i = 0$ to $n - 1$ **do**
    $root \leftarrow uf.find(i)$
    $groups[root].append(i)$
end for
$\hat{\mathcal{C}}^{grouped} \leftarrow [\,]$
for each $group \in groups.values()$ **do**
    $cluster\_group = []$
    for $i \in group$ **do**
        $cluster\_group.append(cluster\_idxs[i])$
    end for
    $\hat{\mathcal{C}}^{grouped}.append(cluster\_group)$
end for
return $\hat{\mathcal{C}}^{grouped}$

Algorithm 3 Group-based Sampling (**GS**)

Require: $\hat{\mathcal{C}}_i^{grouped} = [G_1, G_2, \ldots, G_m]$: grouped mention spans, and k: number of mentions to sample
Ensure: $\hat{c}^{sampled}$: selected mention spans

$m \leftarrow |\hat{\mathcal{C}}^{grouped}|$ {Number of groups}
$n_{list} \leftarrow [|G_i|$ for $G_i \in \hat{\mathcal{C}}^{grouped}]$ {Sizes of each group}
$N \leftarrow \sum n_{list}$ {Total mentions}
if $k > N$ **then**
  error "Cannot sample more mentions than available"
end if
if $k < m$ **then**
  {More groups than needed samples}
  Sort groups in $\hat{\mathcal{C}}^{grouped}$ by size in descending order
  $samples \leftarrow [\,]$
  for $i = 0$ to $k - 1$ **do**
    Randomly select one mention from G_i
    $samples.append(selected\_mention)$
  end for
  return $samples$
end if
{Case when $k \geq m$: allocate at least one sample per group}
$s \leftarrow [1, 1, \ldots, 1]$ {Initialize with one sample per group}
$r \leftarrow k - m$ {Remaining samples to allocate}
if $r > 0$ **then**
  $capacity_i \leftarrow [n_i - 1$ for $n_i \in n_{list}]$ {Remaining capacity per group}
  $total\_capacity \leftarrow \sum capacity_i$
  if $r > total\_capacity$ **then**
    $r \leftarrow total\_capacity$
  end if
  while $r > 0$ **do**
    Find group j with maximum $capacity_j > 0$
    $s[j] \leftarrow s[j] + 1$
    $capacity_j \leftarrow capacity_j - 1$
    $r \leftarrow r - 1$
  end while
end if
$\hat{c}^{sampled} \leftarrow [\,]$
for $i = 0$ to $m - 1$ **do**
  Randomly select $s[i]$ mentions from group G_i
  $\hat{c}^{sampled}.extend(selected\_mentions)$
end for
return $\hat{c}^{sampled}$

results, we report additional experimental results in Tables 8-9 from Appendix D.

Datasets	#Train	#Val	#Test	Avg. W	Avg. M
OntoNotes	2802	343	348	467	56
LitBank	80	10	10	2105	291
WikiCoref	-	-	30	1996	230

Table 7: Dataset statistics: number of documents in each dataset split (#Train/Val/Test), average number of words (i.e., Avg. W) and mentions (i.e., Avg. M) per document.

Metrics. In our experiments, we evaluate performance using the F1 scores of the MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and $CEAF_{\phi_4}$ (Luo, 2005) metrics. The overall CoNLL-F1 score (abbreviated as Avg.F1) is the average of these three F1 scores. To ensure reliability, we report the average for each experiment over 3 different random seeds.

Configurations. In all experiments, unless otherwise specified, we adopt the default settings of $\tau_1 = 50$ and $\tau_2 = 30$ for the Transformer classifier, while maintaining $\tau_1 = 50$ for the linear classifier. When implementing the linear classifier, we follow Xia et al. (2020); Toshniwal et al. (2020); Guo et al. (2023) by representing each historical coreference cluster as a vector representation. This formulation eliminates the necessity of additional parameters beyond the threshold τ_1. We set $\delta = 1 \times 10^{-5}$ and employ the Adafactor optimizer (Shazeer and Stern, 2018) for model training, with learning rates configured as 2e-5 for $\text{DeBERTa}_{\text{large}}$ and 3e-4 for the remaining model layers. All experiments are implemented using the PyTorch-Lightning framework. Each run is executed on a single RTX 4090 GPU with 24GB of VRAM. We have eight such GPUs available, allowing multiple experiments to be conducted concurrently. For fairness, neither data nor model parallelism is employed in any experiment. During training, we accumulate gradients every four steps and set the gradient clipping threshold at 1.0. A linear learning rate scheduler is adopted, incorporating a warm-up phase covering 10% of all training steps. To monitor performance, a validation evaluation is performed every one epoch. The final model is selected based on Avg.F1 in the validation set, with an early stopping patience of 30.

D Additional Experimental Results

In this section, we report additional experimental results, as detailed in Tables 8-9.

Methods	MUC	B^3	$CEAF_{\phi_4}$	Avg.F1
longdoc (Toshniwal et al., 2021)	88.2	75.9	65.5	76.5
Dual-cache (Guo et al., 2023)	88.2	79.2	71.0	79.5
$\text{Maverick}_{\text{mes}}$ (Martinelli et al., 2024)	86.2	78.4	69.6	78.1
$\text{Maverick}_{\text{incr}}$ (Martinelli et al., 2024)	86.5	78.8	69.8	**78.3**
$\text{ImCoref}_{\text{mes}}$ (Luo et al., 2025b)	87.9	79.5	71.6	**79.7**
MEIC-DT(**ours**)	87.9	78.5	77.4	**81.3**
seq2seq (Zhang et al., 2023)	-	-	-	77.3
GPT-3.5-turbo (Le and Ritter, 2023)	-	-	-	75.3
LLMLink (Zhu et al., 2025)	-	-	-	81.5*
$\text{ImCoref-CeS}_{\text{gpt4}}$ (Luo et al., 2025b)	88.8	81.5	72.9	81.1

Table 8: Performance comparison (%) of different methods on LitBank. Note that ($\star$) indicates that it is required to train multiple LLMs to perform coreference task.

Methods	Base Encoders	MUC	B^3	$CEAF_{\phi_4}$	Avg.F1	Params	Training Time	Training Hardware
Supervised neural methods								
c2f-coref (Joshi et al., 2020)	$SpanBERT_{large}$	85.3	78.1	75.3	79.6	370M	-	1× 32G
Icoref (Xia et al., 2020)	$SpanBERT_{large}$	85.3	77.8	75.2	79.4	377M	40h	1×1080TI-12G
CorefQA (Wu et al., 2020)	$SpanBERT_{large}$	88.0	82.2	79.1	83.1$^\diamond$	740M	-	1×TPUv3-128G
s2e-coref (Kirstain et al., 2021)	$LongFormer_{large}$	85.8	79.1	76.1	80.3	494M	-	1× 32G
longdoc (Toshniwal et al., 2021)	$LongFormer_{large}$	85.3	78.0	75.3	79.6	471M	16h	1× A6000-48G
wl-coref (Dobrovolskii, 2021)	$RoBERTa_{large}$	86.3	79.9	76.6	81.0	360M	5h	1×RTX8000-48G
f-coref (Otmazgin et al., 2022a)	DistilRoBERTa	84.4	76.6	74.5	78.5$^\diamond$	91M	-	1×V100-32G
LingMess (Otmazgin et al., 2022b)	$LongFormer_{large}$	86.6	80.5	77.3	81.4	590M	23h	1×V100-32G
Dual-cache (Guo et al., 2023)	$LongFormer_{large}$	86.3	80.3	76.8	81.1	471M	-	1×T4-16G
$Maverick_{mes}$ (Martinelli et al., 2024)	$DeBERTa_{large}$	88.0	82.8	79.9	83.6	504M	14h	1×RTX4090-24G
$Maverick_{incr}$ (Martinelli et al., 2024)	$DeBERTa_{large}$	87.9	82.7	79.8	83.5	452M	29h	1×RTX4090-24G
$ImCoref_{mes}$ (Luo et al., 2025b)	$DeBERTa_{large}$	88.2	83.6	81.0	84.3	531M	13h	1×RTX4090-24G
MEIC-DT(**ours**)	$DeBERTa_{large}$	90.5	84.4	80.2	**85.1**	479M	14h	1×RTX4090-24G
Generative methods								
ASP (Liu et al., 2022)	$FLAN\text{-}T5_{XXL}$	87.2	81.7	78.6	82.5	11B	45h	6×A100-80G
Link-Append (Bohnet et al., 2023)	$mT5_{XXL}$	87.8	82.6	79.5	**83.3**	13B	48h	128×TPUv4-32G
seq2seq (Zhang et al., 2023)	$T0_{XXL}$	87.6	82.4	79.5	83.2	11B	-	8×A100-80G
Large Language Models								
InstructGPT (Le and Ritter, 2023)	API	70.4	58.4	51.7	60.1	-	-	-
		89.2	79.4	73.7	80.8*	-	-	-
GPT-3.5-turbo (Le and Ritter, 2023)	API	66.9	55.5	46.5	56.3	-	-	-
		86.2	79.3	68.3	77.9*	-	-	-
GPT-4 (Le and Ritter, 2023)	API	73.7	62.7	52.3	62.9	-	-	-
		93.7	88.8	82.8	**88.4***	-	-	-
Supervised neural methods+Large Language Models								
$ImCoref\text{-}CeCo_{gpt4o}$ (Luo et al., 2025b)	$DeBERTa_{large}$+API	91.2	84.9	81.8	**86.0**	531M	13h	1×RTX4090-24G

Table 9: Results of different methods on OntoNotes. We report the MUC, B^3, and $CEAF_{\phi_4}$ F1 scores (%), their average (Avg. F1), base encoders, model parameters (Params), and training time/hardware. ($\diamond$) indicates models trained with additional resources; ($\star$) signifies clustering was performed using gold mentions. For the LLMs utilized, we access them remotely via their API interfaces, uniformly setting the temperature parameter to 0.

Methods	MUC	B^3	$CEAF_{\phi_4}$	Avg.F1
longdoc (Toshniwal et al., 2021)	-	-	-	60.1
Dual-cache (Guo et al., 2023)	72.1	62.1	54.7	63.0
LingMess (Otmazgin et al., 2022b)	-	-	-	62.6
$Maverick_{mes}$ (Martinelli et al., 2024)	81.5	65.4	53.5	66.8
$Maverick_{incr}$ (Martinelli et al., 2024)	81.2	65.5	53.7	66.8
$ImCoref_{mes}$ (Luo et al., 2025b)	80.6	66.7	55.6	67.6
MEIC-DT(**ours**)	80.2	68.4	69.4	**72.6**
InstructGPT (Le and Ritter, 2023)	-	-	-	**72.9**
GPT-3.5-turbo (Le and Ritter, 2023)	-	-	-	70.8
$ImCoref\text{-}CeS_{gpt4}$ (Luo et al., 2025b)	83.6	69.8	66.1	**73.2**

Table 10: Performance comparison (%) of different methods on WikiCoref.