scientific reports

Check for updates

OPEN

# A diffusion enhanced CRF and BiLSTM framework for accurate entity recognition

Yunfei Qiu[1], Libo Dong[1,2]✉, Wenwen Zhang[1,2], Haoran Xing[1,2] & Junwei Huang[1,2]

**In Named Entity Recognition tasks, the diffusion model effectively processes discrete data. However, the original model struggles with capturing long-distance dependencies and integrating contextual information, making it difficult to recognize related entities and handle complex syntactic structures. These issues result in ambiguity and uncertainty in entity boundary recognition, affecting overall accuracy and stability. To solve this, we suggest a diffusion model with Conditional Random Fields and Bidirectional Long Short-Term Memory layers. Firstly, the BiLSTM-CRF model captures long-distance dependencies and contextual information, enhancing entity boundary recognition accuracy. Secondly, the Tversky and CRF loss functions select optimal label predictions from the probability distribution, integrating these through weighted summation to enhance sequence dependency processing and label accuracy. Thirdly, we introduce self-attention and graph attention mechanisms to handle complex data structures by processing attention probabilities, integrating with the adjacency matrix, and improving the recognition of key entity relationships. Finally, an automatic noise adjustment mechanism modifies noise levels based on performance, enhancing stability and robustness in inconsistent environments. Experiments demonstrate that this approach improves performance on several NER datasets, with significant gains in recall, accuracy, and F1 scores, making the model more robust in handling noisy and complex environments.**

**Keywords** DIFFUSION+CRF-BiLSTM, Tversky Loss, Graph Attention, Noise Suppression

Named Entity Recognition (NER) refers to detecting and classifying entities into predefined categories, such as persons, locations, and organizations[1]. This task is essential for extracting structured information from unstructured text and supports various natural language processing (NLP) applications, including information retrieval[2], question-answering systems[3], syntactic analysis[4,5], and machine translation[6]. Accurate entity recognition significantly enhances the performance of these tasks[7].

Despite progress in NER, existing models, particularly BiLSTM-CRF, still face significant limitations in complex textual environments, particularly when handling long-range dependencies, noisy data, and nested or overlapping entities. These challenges are especially prominent in real-world scenarios, such as biomedical and legal domains, where sentence structures are more complex and data often contains ambiguities or inconsistencies. These models often struggle to accurately recognize entity boundaries, resulting in reduced performance. For example, BiLSTM-CRF models tend to misidentify boundaries in nested structures or fail to differentiate overlapping entities, which are common in technical and domain-specific texts. In real-world applications, where datasets often exhibit high variability in sentence structures and entity types, the lack of robustness in these models becomes more evident.

Furthermore, although BiLSTM-CRF models capture bidirectional dependencies, they lack mechanisms to explicitly manage uncertainty introduced by noisy data. This issue often causes misclassification, particularly when recognizing nested or overlapping entities, and limits the model's ability to generalize across diverse conditions. Therefore, a model is needed that not only improves entity boundary recognition but also adapts to noisy and complex data while maintaining performance across diverse environments.

Traditional approaches to NER, such as Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs), have contributed significantly to sequence labeling by learning transition probabilities between entities and modeling word dependencies within sentences[8,9]. However, these statistical methods fail to adequately address challenges like nested entities or noisy data, which require more advanced contextual modeling. With the advent of deep learning, BiLSTM models have emerged, enabling the capture of bidirectional contextual

[1]Software Engineering, Liaoning Technical University, Huludao 123105, Liaoning, China. [2]Libo Dong, Wenwen Zhang, Haoran Xing, and Junwei Huang: These authors contributed equally to this work. ✉email: 3066768977@qq.com

information in text[10]. Despite their advantages, BiLSTM models face challenges, especially in handling nested entities, inconsistent data, and the computational complexity of training[11]. Integrating CRF with BiLSTM enhances sequence label consistency[12], but this approach significantly increases computational overhead, reducing scalability for real-time applications.

Recent advances combine pre-trained models like BERT with BiLSTM and CRF to further enhance NER performance[13,14]. These models leverage BERT's contextual understanding and BiLSTM's ability to capture long-term dependencie[15]s, with CRF ensuring sequence label consistency[16,17]. However, the reliance on BERT introduces significant computational costs, making these models less suitable for real-time systems or resource-constrained environments. Furthermore, the lack of mechanisms to address entity-level ambiguities in nested or noisy data often results in suboptimal generalization across datasets. Additionally, generalizing these models across diverse datasets, particularly those containing noisy or non-standard text, remains a significant challenge.

To address these limitations, recent innovations have introduced diffusion models into NLP tasks. Originally developed for image and audio generation[18,19], diffusion models iteratively refine noisy data through forward-reverse processes, offering a unique way to manage uncertainties in textual data. These models introduce and remove Gaussian noise, progressively refining predictions and clarifying complex structures[20,21]. While diffusion models have proven effective in text synthesis and generation tasks, their potential for NER remains underexplored, particularly when integrated with sequence-labeling frameworks.

As shown in Fig. 1, during the forward diffusion process, Gaussian noise is incrementally added to the data (denoted as $x_0 + \varepsilon \sim \mathcal{N}(0,1)$). In the reverse diffusion process (denoted as $X_T$), this noise is progressively removed, sharpening entity boundaries. This process helps the model manage the inherent uncertainty and complexity of textual data, making it particularly effective for handling noisy and ambiguous data[23–25]. However, diffusion models alone still face challenges in consistently handling long-range dependencies and complex syntactic structures, necessitating their integration with other approaches.

To overcome the limitations of current models, we propose a novel approach that integrates the diffusion process with BiLSTM-CRF, enabling more precise boundary recognition and increased robustness in noisy, real-world datasets. The key innovation of this work lies in combining diffusion models' noise management capabilities with BiLSTM's sequential dependency modeling and CRF's structured prediction power. Additionally, we incorporate hybrid attention mechanisms to enhance contextual representations, making the model adept at recognizing nested and overlapping entities in noisy or complex datasets. The diffusion model iteratively refines entity boundaries by adding and removing noise, enabling the model to better handle uncertainty and complexity in text. By integrating BiLSTM's ability to capture long-range dependencies and CRF's sequence labeling capabilities, our model enhances boundary recognition accuracy and overall robustness across various conditions.

The main contributions of the proposed integration of CRF and BiLSTM with diffusion models are as follows:

- **Integration of the diffusion model with BiLSTM-CRF:** We introduce a novel integration where the diffusion model iteratively refines entity boundaries by adding and removing noise, enhancing boundary recognition accuracy in complex, noisy environments. BiLSTM captures long-range contextual dependencies, and CRF ensures global sequence label consistency.
- **Hybrid attention mechanism:** A combination of self-attention and graph attention mechanisms refines contextual embeddings, effectively capturing both local and global dependencies, enabling the recognition of nested and overlapping entities.
- **Innovative loss function design:** The model introduces a multi-task loss function combining Tversky loss and Diffusion loss. Tversky loss optimizes label prediction for imbalanced datasets, while Diffusion loss dy-

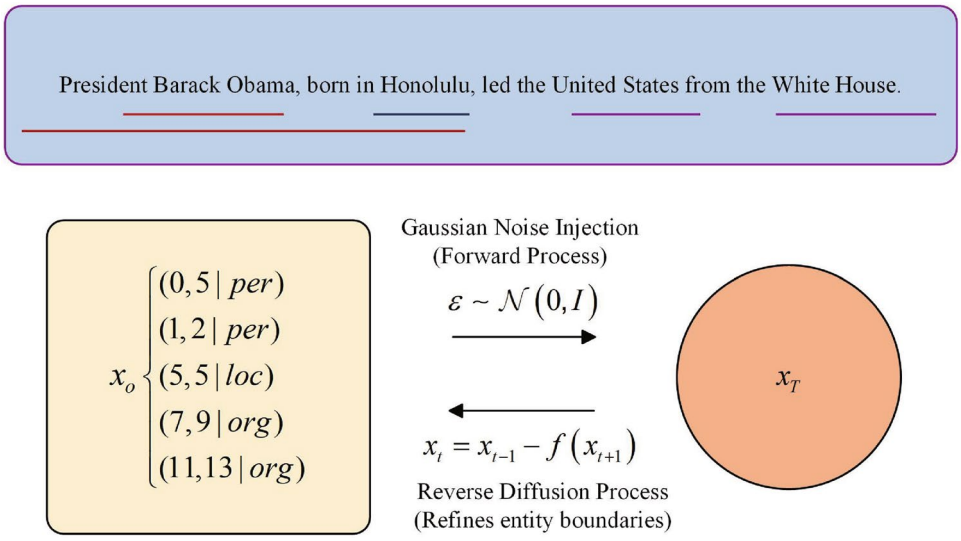


**Fig. 1**. Diffusion-Based NER model overview.

namically adjusts noise levels to prevent gradient vanishing or explosion, improving model stability and generalization.

- **Efficiency optimization and extensive experimental validation:** Optimizing the network architecture and training process reduced the model's training and inference time by 50%. An automatic noise adjustment mechanism was implemented to maintain consistent training throughout the diffusion process. Extensive experiments on both nested and flat NER tasks demonstrate superior performance and computational efficiency compared to state-of-the-art models, underscoring the model's versatility and scalability across various NER tasks.

Experiments demonstrate that the proposed model consistently outperforms existing methods on multiple datasets, including biomedical texts, news articles, and scientific literature, with significant improvements in handling nested and noisy entity recognition.

The structure of the paper is as follows: Section "Related work" reviews related work on NER, covering both traditional and modern approaches. Section "Methods" discusses the construction of the Enhanced Diffusion-CRF-BiLSTM model (EDCBN), including attention mechanisms and the functions used. Section "Algorithm design and complexity" describes the pseudo-algorithm for training and inference. Section "Experimental settings and results" outlines the experimental settings, introduces benchmark datasets, and presents results with a comparative analysis.

## Related work

### Named entity recognition

NER is crucial for information extraction, as it identifies and classifies entities in unstructured text into predefined semantic categories. Traditional NER approaches are broadly divided into annotation-based and span-based methods, each contributing valuable insights to the development of more advanced techniques.

Annotation-based methods, such as CRF and BiLSTM, use sequence tagging to label each text element for entity identification and classification[26,27]. For instance, Ref.[26] combines BiLSTM with CNN to integrate word-level and character-level features, providing a foundation for sequence tagging. However, this method struggles with noisy or diverse text, often misclassifying entities in ambiguous contexts, such as nested or overlapping structures. To address these challenges, robust methods like diffusion-based models have emerged, offering better noise resilience and boundary recognition capabilities. This underscores the need to advance beyond traditional sequence tagging models toward more sophisticated architectures.

Span-based methods identify entity boundaries by predicting their start and end positions in the text[28,29]. Ref.[28] proposed a span-based representation and joint training method for NER, effectively handling overlapping and discontinuous named entities. Despite their precision in detecting flat entity boundaries, these methods exhibit limitations when applied to nested entities or complex sentence structures, often failing in noisy real-world environments. Our proposed diffusion-based approach builds on these strengths by iteratively refining boundary predictions and dynamically handling ambiguous contexts, enabling it to address limitations in existing span-based models. This comparison shows how our model builds on the strengths of span-based methods while addressing their limitations with more dynamic boundary detection techniques.

In recent years, transformer-based models such as BERT and GPT have significantly improved contextual modeling capabilities in NER tasks, establishing themselves as the dominant approach in the field. Through self-attention mechanisms, these models effectively capture long-range dependencies, demonstrating strong performance in tasks involving nested and overlapping entities[30,31]. Ref.[30] proposed a BERT-based NER approach that leverages bidirectional contextual encoding to achieve notable improvements in entity recognition performance across complex domain-specific datasets. However, transformer models often face challenges related to high computational complexity, which limits their scalability in real-time or resource-constrained environments. Moreover, they lack specialized mechanisms for handling nested structures and noisy data, leading to inaccuracies in boundary detection. To address these shortcomings, we propose a diffusion-enhanced BiLSTM-CRF framework, which combines the noise management capabilities of diffusion models with the sequential modeling power of BiLSTM and the global consistency provided by CRF. This results in a lightweight and robust solution for NER tasks.

Large language models (LLMs), such as GPT-3 and GPT-4, have also been applied to NER tasks, leveraging their strong contextual understanding to achieve improved performance in complex scenarios[31]. Ref.[31] demonstrated that GPT-3 excels in identifying ambiguous entities in open-domain tasks due to its advanced contextual reasoning capabilities. However, the study also highlighted significant limitations, including high computational costs and a general-purpose design that make LLMs less effective in tasks requiring rapid inference or handling nested structures. Additionally, LLMs lack task-specific optimizations for sequence labeling, often resulting in suboptimal boundary recognition and sequence consistency. In contrast, our diffusion-enhanced BiLSTM-CRF model is specifically designed for NER tasks, integrating domain-specific optimizations with robust noise suppression mechanisms to achieve superior boundary detection accuracy and computational efficiency.

### Integration of BiLSTM and CRF in the DIFFUSIONNER model

Since 2015, Sohl-Dickstein and colleagues[24] introduced the deep latent generative model, known as the diffusion model, which has significantly advanced image and audio generation[18–20]. This early work laid the foundation for applying diffusion models in various domains; however, the discrete nature of text presents specific challenges for NLP tasks[18,32,33], as explored by Ref.[32]. Text data, unlike continuous signals such as images, requires careful adaptation of diffusion processes to handle its discrete nature. By combining diffusion-based denoising with structured sequence prediction models like BiLSTM-CRF, our approach extends these foundational studies to the NER domain, enabling effective modeling of ambiguous and overlapping entities.

While transformer models effectively capture long-range dependencies through attention mechanisms, diffusion models provide a novel approach by iteratively introducing and removing noise to refine predictions. This makes diffusion models particularly suitable for tasks requiring robustness against noise and complex entity boundaries, as demonstrated in recent NER studies[22]. Unlike transformers, which primarily focus on contextual embeddings, diffusion models excel at explicitly managing uncertainties in the data by progressively refining entity boundaries. This capability is critical for handling nested entities and noisy datasets, where transformers may struggle to maintain consistent predictions. Our integration of diffusion models with BiLSTM and CRF further enhances robustness by combining noise refinement with bidirectional dependency modeling and sequence labeling.

Researchers have explored the application of diffusion models to various text-based tasks, integrating them with additional classifiers such as Diffusion-LM[22], DiffuSeq[21], and SeqDiffuSeq[34]. These methods progressively denoise and refine text generation, extending their applications to complex text structure modeling[35]. This exploration of text generation through diffusion reinforces the relevance of our model in handling the nuanced task of NER, where delineating overlapping or nested entities requires similar progressive refinement techniques. Unlike prior diffusion-based methods that focus on generative tasks, our approach directly integrates diffusion processes into the sequence labeling framework, enabling the iterative refinement of noisy and ambiguous boundary predictions.

In 2023, Shen et al.[23] introduced DiffusionNER, a non-autoregressive diffusion model for entity recognition. This model's success in rapid inference and high performance establishes a strong basis for applying diffusion models to NER, particularly when speed and accuracy are critical in real-time applications. Our work builds on this foundation by integrating BiLSTM and CRF, leveraging their complementary strengths for contextual modeling and sequence consistency. Additionally, our use of hybrid attention mechanisms further enhances the model's ability to capture both local and global dependencies, providing significant advantages over prior diffusion-based NER approaches.

# Methods

## Model overview

The Enhanced Diffusion Boundary Classification Network (EDCBN) addresses key challenges in NER, including noisy annotations, overlapping entities, and complex text structures, through a unified framework designed to enhance feature representation, refine boundary predictions, and ensure structured labeling. The architecture integrates five core modules: BERT for contextual embeddings, BiLSTM for sequence modeling, attention mechanisms (self-attention and graph attention) for feature refinement, a diffusion process for boundary denoising, and a CRF layer for structured sequence prediction.

Each component of this framework contributes uniquely while complementing the others. BERT captures local and global semantics but lacks label dependency modeling, which is resolved by the CRF layer ensuring global sequence consistency. BiLSTM models forward and backward dependencies but struggles with noisy data, a limitation addressed by the diffusion process that iteratively refines predictions. The attention mechanisms focus on token-level and entity-level interactions, strengthening feature representation and enabling the model to effectively handle ambiguous and overlapping boundaries. Together, these components form a robust architecture designed to address the inherent complexities of NER tasks.

Figure 2 illustrates the EDCBN architecture. The process begins with BERT generating token-level contextual embeddings from input text. These embeddings are passed through a BiLSTM layer to model sequential dependencies, followed by self-attention and graph attention mechanisms to capture both local interactions and broader contextual relationships. The CRF layer ensures coherent and globally consistent label sequences, while the diffusion mechanism iteratively introduces and removes noise to refine entity boundaries and improve robustness.

This integrated framework provides a targeted solution to key NER challenges. By iteratively refining entity boundaries, the diffusion mechanism addresses noisy data and boundary ambiguities. The CRF layer, in tandem with the diffusion process, improves precision and robustness, especially for overlapping and nested entities. Hybrid attention mechanisms enhance the model's ability to capture both local and global dependencies, essential for nested structures. Together, these components enable EDCBN to achieve superior performance on noisy datasets, handle complex entity structures, and generalize effectively across diverse text types.

## Input and sequence modeling

The EDCBN transforms input text into structured predictions through a pipeline that involves contextual embeddings, feature refinement via attention mechanisms, and hierarchical sequence modeling. This process is illustrated in Fig. 3, which provides a detailed flowchart of the EDCBN model's architecture, outlining the interactions between its various components and their contributions to the structured prediction task.

*BERT for contextual embedding*

The input sequence $S = x_1, x_2, \ldots, x_n$, where $n$ represents the number of tokens, is first passed through BERT, a pre-trained bidirectional transformer. BERT generates token-level contextual embeddings $E$, which serve as the foundation for subsequent refinement.

***Mathematical Representation***

The BERT embeddings are computed as:

$$E = \mathrm{BERT}(S) \tag{1}$$

where $E \in \mathbb{R}^{n \times d}$ represents the embeddings for $n$ tokens, with $d$ denoting the embedding dimension.

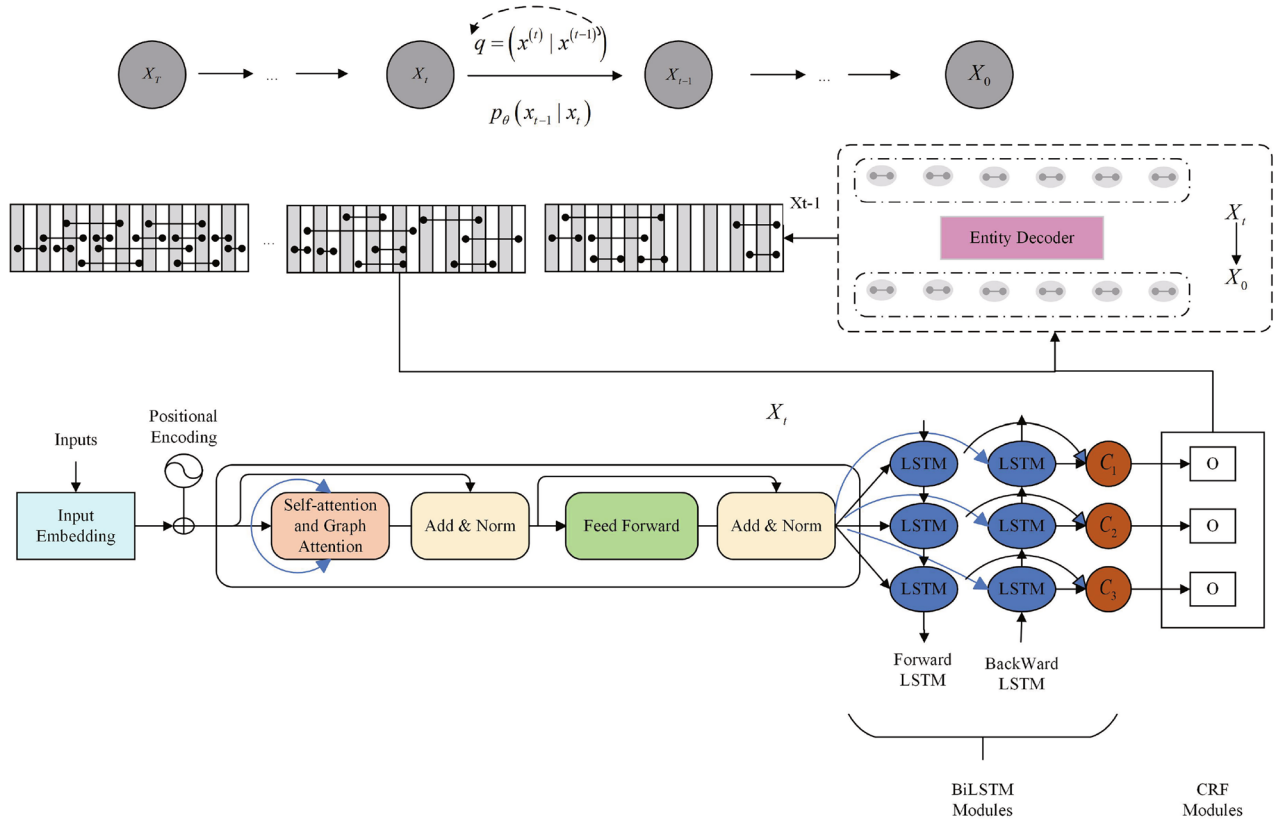


**Fig. 2**. Comprehensive framework diagram: BERT-BiLSTM-CRF with diffusion for enhanced entity recognition.

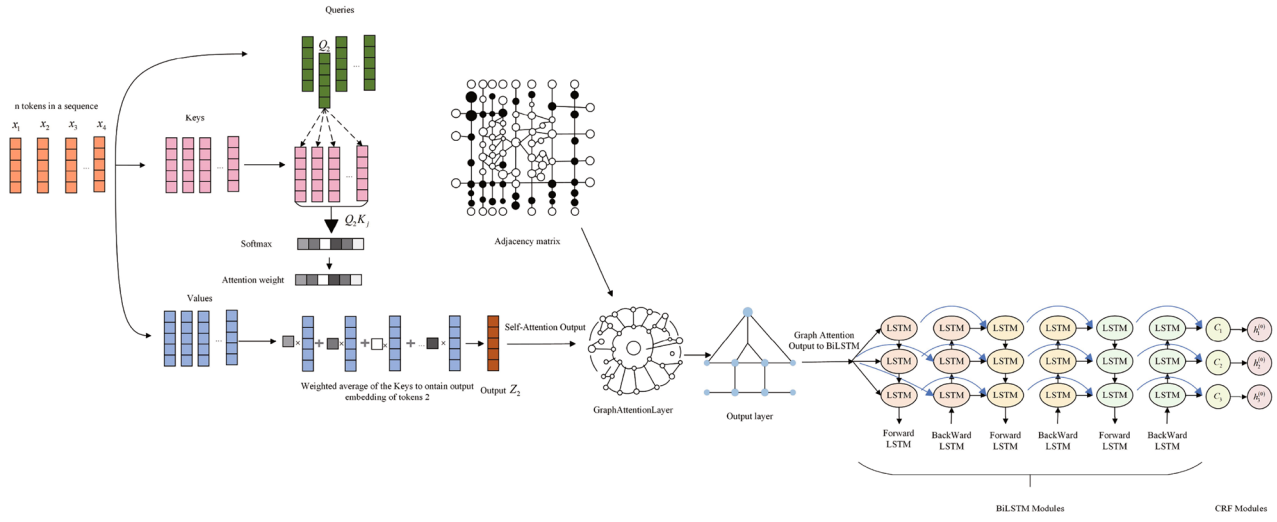


**Fig. 3**. Workflow of the EDCBN model: step-by-step transformation from input to structured prediction.

BERT provides dynamic token embeddings that encode contextual information by capturing both local and global dependencies within the sequence. This capability is particularly beneficial for NER tasks, as it facilitates the resolution of polysemous words, entity boundary ambiguities, and long-range dependencies. The generated embeddings are subsequently refined by the Hybrid Attention Mechanism.

*Hybrid attention mechanism for feature refinement*

The Hybrid Attention Mechanism combines Self-Attention and Graph Attention to refine the BERT embeddings, enabling the model to capture both local (token-level) and global (entity-level) dependencies in a unified representation.

***Self-Attention***

Self-Attention computes token-to-token dependencies within the sequence, focusing on local interactions. For each token embedding *E*, it calculates:

$$Q = W_Q E, \quad K = W_K E, \quad V = W_V E \tag{2}$$

$$\text{Self-Attention}(E) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{3}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ are learnable weight matrices, and $d_k$ is the dimension of the keys.

***Graph Attention***

Graph Attention models relationships between tokens as nodes in a graph, leveraging an adjacency matrix Adj to capture global dependencies. The graph attention is computed as:

$$\hat{A} = \sigma(W_A(A \odot \text{Adj}) + b_A) \tag{4}$$

$$Z = \text{concat}(A_1V, A_2V, \ldots, A_hV)W_O \tag{5}$$

where $W_A$ and $b_A$ are learnable parameters, $A$ is the node feature matrix, and $W_O$ is a projection matrix.

***Fusion of attention mechanisms***

The outputs of Self-Attention and Graph Attention are integrated through a fusion mechanism, defined as:

$$\text{Hybrid-Attention}(E) = \gamma_1 \cdot \text{Self-Attention}(E) + \gamma_2 \cdot \text{Graph-Attention}(E) \tag{6}$$

where $\gamma_1$ and $\gamma_2$ are learnable weights that dynamically balance the contributions of the two attention mechanisms. Initially, these weights are set to $\gamma_1 = \gamma_2 = 0.5$ to give equal importance to both mechanisms. During training, they are optimized via backpropagation.

To ensure stability, the weights can be normalized using:

$$\gamma_1 + \gamma_2 = 1 \tag{7}$$

or, alternatively, softmax normalization:

$$\gamma_1 = \frac{\exp(a_1)}{\exp(a_1) + \exp(a_2)}, \quad \gamma_2 = \frac{\exp(a_2)}{\exp(a_1) + \exp(a_2)} \tag{8}$$

***Output of the hybrid attention mechanism***

The refined embeddings are denoted as:

$$E_{\text{refined}} = \text{Hybrid-Attention}(E) \tag{9}$$

These embeddings are passed to the BiLSTM-CRF module for sequence modeling and structured prediction.

*Fused BiLSTM-CRF module*

The BiLSTM-CRF module processes the refined embeddings $E_{\text{refined}}$ to extract hierarchical sequence features and ensure globally consistent predictions.

***Three-layer BiLSTM***

The BiLSTM module consists of three stacked layers, where each layer captures dependencies at different levels of abstraction:

- **Layer 1**: Captures short-range dependencies between adjacent tokens.
- **Layer 2**: Models mid-range interactions, such as phrase-level relationships.
- **Layer 3**: Integrates long-range dependencies across the sequence.

For the *i*-th layer ($i = 1, 2, 3$), the forward and backward passes are computed as:

$$H^{(i)}_{\text{forward}} = \overrightarrow{\text{LSTM}}(H^{(i-1)}), \quad H^{(i)}_{\text{backward}} = \overleftarrow{\text{LSTM}}(H^{(i-1)}) \tag{10}$$

$$H^{(i)} = H^{(i)}_{\text{forward}} \oplus H^{(i)}_{\text{backward}} \tag{11}$$

where $H^{(0)} = E_{\text{refined}}$.

***CRF for structured prediction***

The final BiLSTM output $H^{(3)}$ is passed to the CRF layer, which predicts globally consistent label sequences by modeling dependencies between adjacent labels. The conditional probability of a label sequence *Y* is given by:

$$P(Y|H) = \frac{\exp\left(\sum_{i=1}^{n} \phi(y_{i-1}, y_i, H_i)\right)}{\sum_{Y' \in \mathcal{Y}} \exp\left(\sum_{i=1}^{n} \phi(y'_{i-1}, y'i, H_i)\right)} \tag{12}$$

where $\phi(yi-1, y_i, H_i)$ represents the transition and emission scores.

***Integration of BiLSTM and CRF***

The hierarchical features extracted by BiLSTM are structured into valid label sequences by the CRF layer, ensuring that both contextual and sequential dependencies are effectively modeled.

## Diffusion-based refinement process

The diffusion-based refinement process in the EDCBN framework, shown in Fig. 4, resolves boundary ambiguities and noisy annotations by iteratively refining sequence representations with Gaussian noise and dynamic noise adjustment, ensuring robust and consistent predictions.

*Input to the diffusion process*

The structured sequence representations generated by the CRF layer serve as the input to the diffusion process. These representations are defined as:

$$H^{(0)} = \{h_1^{(0)}, h_2^{(0)}, \ldots, h_T^{(0)}\} \tag{13}$$

where each token-level feature vector $h_t^{(0)}$ belongs to a $d$-dimensional space:

$$h_t^{(0)} \in \mathbb{R}^d \tag{14}$$

These feature vectors encode both contextual and sequential dependencies, capturing local relationships and global sequence structure. As outputs of the CRF layer, these representations are optimized for structured prediction tasks, providing a robust foundation for the subsequent refinement in the diffusion process.

*Forward diffusion with automatic noise adjustment*

In the forward diffusion phase, Gaussian noise is progressively injected into the structured sequence representations to simulate boundary uncertainties. This step introduces controlled perturbations, allowing the

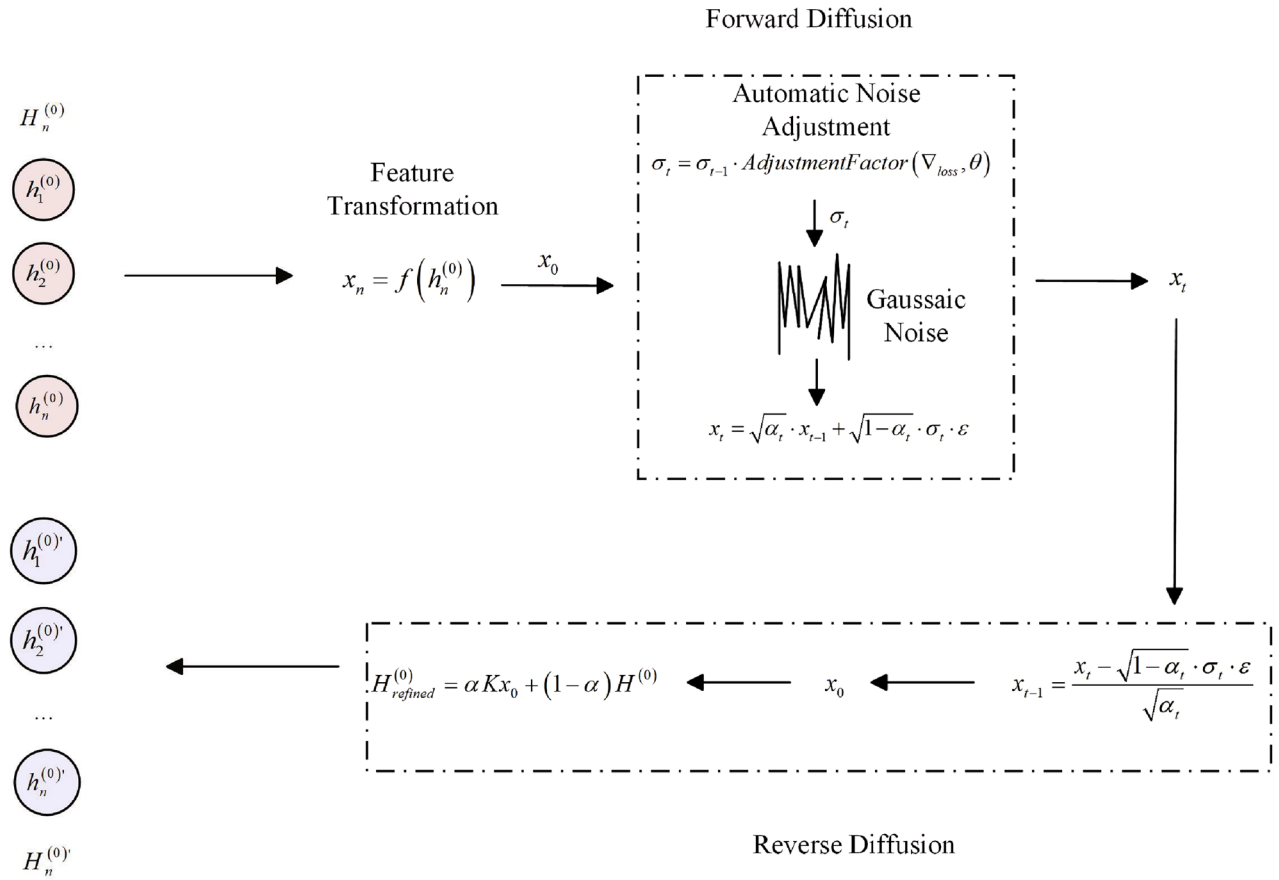


**Fig. 4**. Diffusion-based refinement process in EDCBN.

model to explore variations in entity boundaries and account for potential ambiguities in noisy or inconsistent datasets. At each diffusion step $t$, the sequence representation $H^{(t)}$ is updated as:

$$x_t = \sqrt{\alpha_t} \cdot x_{t-1} + \sqrt{1-\alpha_t} \cdot \sigma_t \cdot \varepsilon \tag{15}$$

where $\alpha_t \in (0,1)$ controls the balance between retaining the original representation and introducing noise, $\sigma_t$ represents the noise scale, and $\varepsilon \sim \mathcal{N}(0, I)$ denotes Gaussian noise sampled from a standard normal distribution.

Gaussian noise is selected for its desirable mathematical properties, including symmetry, zero-mean characteristics, and smooth continuity. These attributes ensure unbiased perturbations and allow controlled transformations of the feature space. In the context of Named Entity Recognition (NER), Gaussian noise effectively simulates the inherent uncertainties in boundary predictions caused by noisy annotations or ambiguous text structures. This process acts as a regularization mechanism, mitigating overfitting and improving the model's robustness to diverse input scenarios.

The automatic noise adjustment mechanism further optimizes this phase by dynamically scaling the noise level $\sigma_t$ based on the training state of the model. This mechanism adjusts the balance between noise injection and structural preservation, ensuring that the forward diffusion phase remains effective across different training conditions. The adjusted noise scale is computed as:

$$\sigma_{t+1} = \sigma_t \cdot f(\nabla_{\text{loss}}, \theta) \tag{16}$$

where $\nabla_{\text{loss}}$ is the gradient of the loss function and $\theta$ represents the model parameters. By incorporating feedback from the training process, the noise adjustment mechanism dynamically adapts to the learning capacity of the model, allowing for stable and efficient training.

The output of the forward diffusion phase is a noisy sequence representation:

$$H^{(N)}_{\text{noisy}} \tag{17}$$

where $H^{(N)}_{\text{noisy}}$ captures a robust approximation of the input features, enriched with boundary variations introduced by Gaussian noise. These representations serve as the input to the reverse diffusion phase for iterative refinement.

***Automatic noise adjustment mechanism***

To further optimize the diffusion process, an automatic noise adjustment mechanism dynamically adapts the noise scale $\sigma_t$ based on the model's training state. This mechanism ensures that the noise injection remains proportional to the model's learning capacity at each stage, striking a balance between feature exploration and structural preservation. The adjustment is computed as:

$$\sigma_{t+1} = \sigma_t \cdot \text{AdjustmentFactor}(\nabla_{\text{loss}}, \theta) \tag{18}$$

where $\nabla_{\text{loss}}$ is the gradient of the loss function, and $\theta$ represents the model parameters. During training, this mechanism dynamically scales $\sigma_t$ to either increase or decrease noise injection based on the model's learning state, enabling the model to explore boundary variations effectively without destabilizing the feature representations.

*Reverse diffusion phase: iterative feature refinement*

The reverse diffusion phase reconstructs the noisy sequence representations by iteratively removing the injected Gaussian noise. This phase is defined as:

$$x_{t-1} = \frac{x_t - \sqrt{1-\alpha_t} \cdot \sigma_t \cdot \varepsilon}{\sqrt{\alpha_t}} \tag{19}$$

where the noise scale $\sigma_t$ and perturbations $\varepsilon$ are consistent with the forward phase to ensure alignment between the two processes.

The learned diffusion kernel $K$, a $T \times T$ matrix, further refines the features by propagating information across tokens based on sequence-level similarities or structural proximity:

$$H^{(k+1)} = \alpha \cdot K \cdot H^{(k)} + (1-\alpha) \cdot H^{(0)} \tag{20}$$

where $H^{(k)}$ represents the sequence features at the $k$-th iteration. This iterative refinement process continues until convergence or a predefined number of iterations, producing the final sequence representation:

$$H^{(0)}_{\text{refined}} = \alpha K x_0 + (1-\alpha) H^{(0)} \tag{21}$$

The diffusion mechanism is an adaptive refinement process fully integrated into the EDCBN architecture. In the forward diffusion phase, Gaussian noise is injected into the structured sequence predictions generated by the CRF layer, simulating boundary ambiguities. This process is dynamically optimized through the automatic noise adjustment mechanism, which scales the noise injection based on the model's learning state. The reverse diffusion phase then iteratively denoises these representations, recovering refined features that ensure both local consistency and global coherence. By seamlessly combining noise injection and iterative refinement, the

diffusion mechanism enhances the EDCBN framework, addressing noisy annotations and boundary ambiguities while improving robustness and accuracy in challenging NER tasks.

## Optimized advanced loss functions

In this research, we enhance the performance of the NER model by integrating a suite of advanced loss functions, each tailored to address specific challenges in the learning process. These loss functions work in tandem to handle data imbalance, maintain sequence integrity, and improve boundary prediction accuracy. Their combination allows the model to achieve robust and generalizable performance across diverse NER tasks.

*Tversky loss function*

The Tversky loss function is employed to tackle the issue of class imbalance, which is common in NER datasets where some entity types appear less frequently than others. By generalizing the Dice coefficient, the Tversky loss introduces tunable parameters $\alpha$ and $\beta$ to control the penalties for false positives and false negatives, making it particularly effective in imbalanced scenarios. The loss is defined as:

$$L_{Tversky}(Y, \hat{Y}) = 1 - \frac{\sum_i Y_i \hat{Y}_i + \epsilon}{\sum_i Y_i \hat{Y}_i + \alpha \sum_i Y_i (1 - \hat{Y}_i) + \beta \sum_i (1 - Y_i) \hat{Y}_i + \epsilon} \tag{22}$$

where $Y_i$ denotes the true probability, $\hat{Y}_i$ is the predicted probability, and $\epsilon$ ensures numerical stability. The parameters $\alpha$ and $\beta$ are adjusted to prioritize precision or recall depending on the dataset characteristics. For instance, increasing $\alpha$ penalizes false negatives more heavily, which is crucial for identifying rare entity types.

*CRF loss function*

To ensure sequence-level consistency and accurately model dependencies between adjacent labels, we incorporate the CRF loss function. This loss is critical for maintaining structured predictions in NER, where label transitions (e.g., "B-PER" to "I-PER") must adhere to specific rules. The CRF loss is defined as:

$$L_{CRF}(Y, H) = -\log \left( \frac{\exp\left(\sum_i \phi(y_{i-1}, y_i, H_i)\right)}{\sum_{Y' \in \mathcal{Y}} \exp\left(\sum_i \phi(y'_{i-1}, y'_i, H_i)\right)} \right) \tag{23}$$

where $\phi(y_{i-1}, y_i, H_i)$ represents the transition and emission scores, and $\mathcal{Y}$ is the set of all possible label sequences. By maximizing the probability of the correct sequence, this loss ensures that the model produces coherent and valid predictions across tokens.

*Boundary loss*

Boundary prediction is a critical challenge in NER, particularly for complex datasets with overlapping or nested entities. To improve the model's ability to detect precise entity boundaries, we introduce a Boundary Loss function, which directly optimizes the alignment between predicted and true boundary markers. The loss is computed as:

$$L_{boundary}(Y, \hat{Y}) = \sum_i |Y_{boundary_i} - \hat{Y}_{boundary_i}| \tag{24}$$

where $Y_{boundary_i}$ and $\hat{Y}_{boundary_i}$ denote the true and predicted boundary values, respectively. By focusing on boundary-specific errors, this loss complements the CRF loss, which prioritizes sequence-level consistency, ensuring that both entity boundaries and overall label sequences are accurately modeled.

*Combined loss function*

To balance the contributions of the individual loss functions, we define a combined loss function as a weighted sum:

$$L_{total} = \lambda_1 L_{Tversky} + \lambda_2 L_{CRF} + \lambda_3 L_{boundary} + \lambda_4 L_{cross\_entropy} \tag{25}$$

where $\lambda_1, \lambda_2, \lambda_3$, and $\lambda_4$ are weighting factors that control the relative importance of each loss component. These weights can be tuned to optimize the model's performance on specific datasets or tasks. For our low-resource NER task, we conducted systematic experiments with different weight configurations. Based on validation results, we found that setting $\lambda_1 = 1.0$, $\lambda_2 = 0.7$, $\lambda_3 = 1.2$, $\lambda_4 = 0.8$ yields optimal performance. While maintaining a standard weight for Tversky loss ($\lambda_1 = 1.0$) to address data imbalance, we slightly reduced the CRF loss weight ($\lambda_2 = 0.7$) to prevent overfitting to sequence patterns in limited data scenarios. We enhanced the boundary detection loss ($\lambda_3 = 1.2$) to improve precise entity identification, which is particularly challenging in low-resource contexts. Meanwhile, we moderately reduced the cross-entropy loss weight ($\lambda_4 = 0.8$) to balance the overall learning objective.

The integration of these loss functions enhances the model's robustness by addressing different aspects of the learning process. Tversky Loss mitigates the impact of data imbalance, ensuring rare entities are correctly identified. CRF Loss maintains sequence-level integrity, enabling coherent label predictions. Boundary Loss focuses on precise entity boundary detection, which is crucial for complex NER scenarios. Together, these

loss functions align the model's predictions closely with the true probability distributions, improving overall accuracy and generalization across diverse datasets.

## Algorithm design and complexity

This section presents the implementation details and complexity analysis of the algorithms within the EDCBN framework. It describes the training and inference processes, emphasizing parameter optimization and model operation during inference.

### Training algorithm

The training of the EDCBN model iteratively updates parameters to minimize the loss function. It combines diffusion-based boundary refinement, noise injection, and gradient optimization. The model employs BERT for token embeddings, followed by BiLSTM and CRF for structured prediction. Algorithm 1 outlines the training steps for the EDCBN model.

**Require:** $\mathcal{D}$ - input dataset, BERT model for embedding extraction
**Ensure:** $\theta$ - optimized model parameters
Initialize model parameters $\theta$
**while** each epoch until convergence **do**
  **for** each $(S)$ in $\mathcal{D}$ **do** ▷ Iterate over all sentences $S$ in dataset $\mathcal{D}$
    Compute token embeddings $E = \text{BERT}(S)$ ▷ Obtain embeddings using BERT model
    Normalize $E$ to $K$ boundaries ▷ Apply boundary normalization based on token embeddings
    Set initial boundary representation $x_0$
    **for** $t = 1$ to $T$ **do**
      Inject Gaussian noise $\mathcal{N}(0, I)$ into $x_{t-1}$
      Update boundary representation $x_t$ via forward diffusion
    **end for**
    Compute class probabilities and boundary predictions ▷ Using BiLSTM and CRF layers
    Update model parameters $\theta$ using backpropagation and gradient descent ▷ Optimize with composite loss (Tversky + CRF)
  **end for**
**end while**
**return** $\theta$ ▷ Return the optimized parameters after training

**Algorithm 1**. Training EDCBN model.

**Explanation of Algorithm 1**

In this training process, each sentence $S$ is processed individually to obtain its token embeddings using the BERT model ($E = \text{BERT}(S)$). The embeddings are then normalized to establish the initial boundary representation $x_0$. Gaussian noise is incrementally introduced into the boundary representation at every time step throughout the forward diffusion.

At each time step during the forward diffusion process" ? "at every time step throughout the forward diffusion The final boundary and class predictions are computed through the BiLSTM and CRF layers, with model parameters $\theta$ updated using backpropagation concerning a composite loss function that combines Tversky loss for boundary prediction and CRF loss for sequence labeling.

Iterating systematically through all sentences ensures that every sentence $S$ in the dataset $\mathcal{D}$ contributes to the model's learning.

### Inference algorithm

The inference process utilizes the trained EDCBN model to predict entity boundaries and labels on new data. This process uses the trained model parameters $\theta$ and employs self-attention and graph attention mechanisms to iteratively refine the entity predictions. Algorithm 2 describes the inference

| Model Variant | Time Complexity | Training Time (h) | Inference Time (ms) |
|---|---|---|---|
| Diffusion | $O(T \cdot n \cdot d)$ | 4 | 120 |
| + BiLSTM | $O(T \cdot n \cdot d) + O(L \cdot n \cdot d^2)$ | 5 | 180 |
| + CRF | $O(n \cdot k^2)$ | 5.5 | 190 |
| + Graph Attention | $O(n^2 \cdot d)$ | 6 | 240 |
| + Self-Attention | $O(n^2 \cdot d)$ | 6 | 310 |

**Table 1**. Time Complexity and Performance of EDCBN Components (ACE2004).

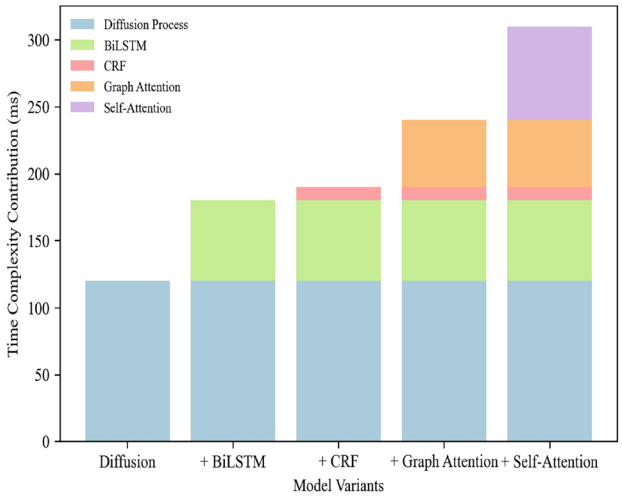


**Fig. 5**. Time complexity contributions of EDCBN components.

**Require:** $T$ - set of diffusion steps, $\theta$ - trained model parameters
**Ensure:** Set of recognized entities
1: Initialize noisy boundary representation $x_T$ with Gaussian noise
2: **for** $t = T$ down to 1 **do**
3: Apply denoising function $f_\theta(x_t)$ to refine boundaries ▷ Denoising based on learned model parameters $\theta$
4: Recover boundary representation $x_{t-1}$ from $x_t$
5: **end for**
6: Identify likely entity boundaries from $\hat{x}_0$ ▷ Extract entities from denoised boundary representation
7: Refine entity boundaries and resolve any overlaps
8: **return** Set of recognized entities

**Algorithm 2**. Inference for EDCBN.

**Explanation of Algorithm 2**

In the inference process, the noisy boundary representation $x_T$ is progressively refined using the trained model's parameters $\theta$ through a denoising function $f_\theta(x_t)$. The denoising function $f_\theta$is a learned function that models the reverse diffusion process, gradually removing the Gaussian noise and recovering the clean boundary representation.

The process iteratively recovers less noisy boundary representations $x_{t-1}$ until the clean boundary representation $\hat{x}_0$ is obtained. From this final representation, entity boundaries are identified, refined, and any overlapping entities are resolved.

## Algorithm design and complexity

In this section, the computational complexity of the EDCBN framework is comprehensively analyzed. The framework is decomposed into its key components, their respective time complexities are analyzed, and the overall computational cost is presented. We address reviewers' comments on the importance of explicitly

accounting for complexity during training and inference and discuss the impact of added modules such as BiLSTM, CRF, graph attention, and self-attention. Complexity contributions are summarized in Table 1, and their proportional impacts are visualized in Fig. 5.

The following sections detail the time complexity of each component in the EDCBN framework, and Fig. 5 visually summarizes how the complexity contributions from each component grow as additional modules are introduced:

**Diffusion process:** The diffusion process operates in both forward and reverse directions. Each step processes a sequence of length $n$ with an embedding dimension $d$. With $T$ diffusion steps, the complexity is given by:

$$\text{Forward diffusion: } O(T \cdot n \cdot d), \quad \text{Reverse diffusion: } O(T \cdot n \cdot d).$$

**BiLSTM layers:** Each BiLSTM layer processes the sequence bidirectionally, with a complexity of $O(n \cdot d^2)$ per layer. For $L$ BiLSTM layers, the total complexity is:

$$O(L \cdot n \cdot d^2),$$

where $L = 3$ in our framework.

**CRF decoding:** The CRF layer performs dynamic programming over $n$ tokens and $k$ labels to compute the most probable sequence. The complexity is:

$$O(n \cdot k^2).$$

**Graph attention mechanism:** The graph attention mechanism computes pairwise relationships between $n$ nodes in the adjacency matrix. The resulting complexity is:

$$O(n^2 \cdot d).$$

**Self-attention mechanism:** Self-attention involves comparing all token pairs within a sequence. The complexity is:

$$O(n^2 \cdot d).$$

**Total complexity:** Combining all components, the overall complexity of the EDCBN framework is:

$$O(T \cdot n \cdot d) + O(L \cdot n \cdot d^2) + O(n \cdot k^2) + O(n^2 \cdot d).$$

The EDCBN framework's computational cost grows as additional modules (BiLSTM, CRF, graph attention, and self-attention) are introduced. As illustrated in Fig. 5, the contributions of each component to the overall time complexity are visually summarized. The diffusion process dominates with $O(T \cdot n \cdot d)$, particularly for longer sequences or larger embedding dimensions. Graph and self-attention contribute significantly to the quadratic growth in sequence length $n$ due to pairwise comparisons. However, these modules provide essential global context modeling, leading to performance gains.

## Experimental settings and results

### Datasets

In this study, we rigorously evaluated the performance of our proposed NER model using five diverse and specialized datasets: ACE2004[36], GENIA[37], CoNLL2003[38], SciERC[39], and NCBI-Disease[40]. These datasets were selected for their representation of diverse NER challenges across different domains, enabling a thorough assessment of the model's robustness, versatility, and generalization to various types of text. The specific characteristics of each dataset are outlined below:

- **ACE2004:** This dataset focuses on nested entity recognition within news texts. The primary challenge of this dataset lies in its complex, multi-layered entity structure, where entities are often embedded within other entities. Correctly identifying these nested structures is crucial for enhancing the model's ability to handle real-world texts with intricate syntactic dependencies. We selected this dataset to evaluate the model's effectiveness in recognizing complex relationships between entities.
- **GENIA:** The GENIA dataset is sourced from the biomedical domain and comprises highly specialized biological terms. This dataset poses challenges due to its domain-specific terminology and long-distance dependencies within the text. It assesses the model's ability to generalize to niche scientific areas where linguistic patterns may differ significantly from general language use. Selecting GENIA enables us to evaluate the model's performance on domain-specific NER tasks, particularly in the life sciences.
- **CoNLL2003:** As one of the most widely used flat NER benchmarks, the CoNLL2003 dataset comprises news articles containing person, location, and organization entities. It serves as a standard testbed for NER models, enabling us to compare our results against various state-of-the-art methods. The primary motivation for including this dataset is to evaluate the model's generalization ability for standard NER tasks in general news texts, which feature simpler, non-nested entity structures compared to ACE2004.
- **SciERC:** The SciERC dataset consists of scientific papers from the computer science domain and presents the challenge of recognizing domain-specific terms and identifying relationships among them. The dataset evaluates the model's ability to perform NER on highly technical content, where the distinction between sim-

ilar terms may be subtle. It also assists in evaluating the model's performance in extracting entities and their relationships in academic papers, which necessitate a strong understanding of technical terminology.
- **NCBI-Disease:** This dataset, sourced from the biomedical domain, focuses specifically on disease-related entity recognition. One of the key challenges in this dataset is the high level of noise present in biomedical texts, such as inconsistent naming conventions and acronyms. Including NCBI-Disease evaluates the model's ability to handle noisy, domain-specific datasets and its robustness in recognizing medical terminology.

In summary, the selection of these datasets ensures a comprehensive evaluation across multiple domains while highlighting the specific strengths of the proposed model. The nested entities in ACE2004 evaluate the model's capacity to handle complex entity structures and long-distance dependencies. GENIA and NCBI-Disease present domain-specific challenges that evaluate the model's ability to generalize to highly specialized biomedical vocabularies. SciERC, with its technical terms in the computer science domain, tests the model's precision in distinguishing between similar technical entities, while CoNLL2003, as a standard NER benchmark, allows for direct comparison with existing methods, showcasing the model's versatility across both standard and domain-specific tasks. Collectively, these datasets ensure that the model is rigorously tested for generalizability, robustness, and accuracy across various applications.

## Implementation details

We employed precision, recall, and F1 score as evaluation criteria, calculated at the entity and token levels to guarantee a thorough evaluation of the model's performance. These metrics, widely used in NER tasks, offer a balanced measure of model accuracy and stability across different datasets. The calculation methods are as follows:

- **Precision:**

$$\text{Precision} = \frac{TP}{TP + FP} \tag{26}$$

where *TP* denotes the number of true positives (correctly predicted entities) and *FP* denotes the number of false positives (incorrectly predicted entities).

- **Recall:**

$$\text{Recall} = \frac{TP}{TP + FN} \tag{27}$$

where *FN* denotes the number of false negatives (entities that the model missed).

- **F1 score:**

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{28}$$

The F1 score is the harmonic mean of precision and recall, offering a balanced measure between the two. It is particularly useful for assessing the trade-off between precision and recall, ensuring that both over-prediction and under-prediction of entities are taken into account.

Experiments were performed using an NVIDIA GeForce RTX 3090 GPU (24 GB VRAM), with training durations varying across datasets depending on batch size and complexity. On average, training required X hours per dataset. The model was trained with the Adam optimizer, and a linear warm-up and decay strategy was applied to the learning rate to stabilize training, particularly in the early stages.

For text encoding, we employed bert-large and biobert-large models, which were fine-tuned to meet the specific requirements of each dataset. These pre-trained models offer strong baseline representations, which we adapted for the NER task by introducing task-specific layers (i.e., BiLSTM and CRF layers). Fine-tuning was conducted by adjusting the learning rates for the pre-trained and task-specific layers separately, with the learning rate for the pre-trained layers set lower to preserve the knowledge acquired from large-scale corpora.

Batch sizes were optimized for each dataset to ensure stable gradient updates and efficient use of GPU memory. Specifically, for ACE2004, CoNLL2003, SciERC, and NCBI, a batch size of 8 was used, while a batch size of 4 was chosen for GENIA, due to the complexity and length of its input sequences. This variation in batch sizes was determined empirically by monitoring convergence rates and the stability of the training process across datasets.

The model architecture included three BiLSTM layers and a CRF layer for entity boundary detection, with each layer contributing to the model's ability to capture long-distance dependencies and ensure consistent sequence labeling. The CRF layer enables the model to learn optimal label sequences by considering the dependencies between adjacent labels.

To further enhance the model's robustness, we conducted hyperparameter tuning through grid search, experimenting with various settings for learning rates, batch sizes, and regularization techniques. The final hyperparameter choices were those that consistently achieved the best performance across validation sets for all datasets. Table 2 summarizes the key parameters used during training and evaluation.

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| train log iter | 1 | split epoch | 10 |
| train batch size | 8 | stage one lr scale | 2.0 |
| epochs | 100 | prop drop | 0.3 |
| lr | 5e-05 | num proposals | 60 |
| lr warmup | 0.1 | sampling timesteps | 5 |
| weight decay | 0.01 | timesteps | 1000 |
| max grad norm | 1.0 | scale | 1.0 |
| match boundary weight | 1.0 | eval batch size | 16 |
| match class weight | 1.0 | entity threshold | 2.5 |
| loss boundary weight | 1.2 | seed | 488 |
| loss class weight | 0.8 | BiLSTM layers | 3 |
| loss tversky weight | 1.0 | CRF layer | 1 |
| loss crf weight | 0.7 | Diffusion time steps (T) | 1000 |
| repeat gt entities | 60 | Noise intervals (K) | 60 |
| eval every epochs | 1 | Sampling interval ($\gamma$) | 5 |
| sampling processes | 4 | | |

**Table 2**. Model training and evaluation parameters.

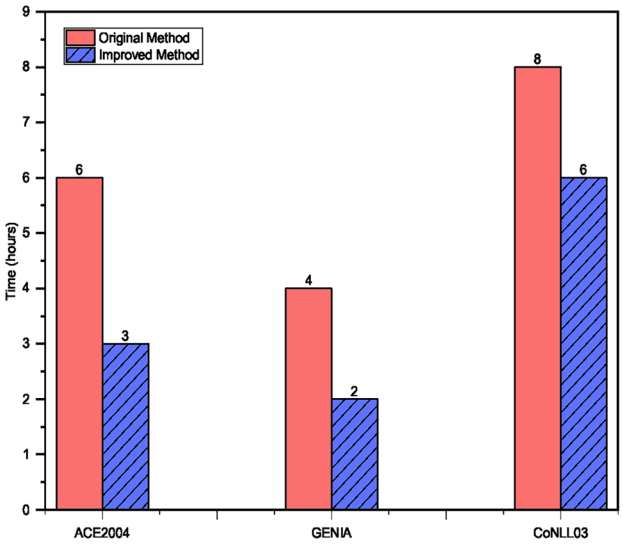


**Fig. 6**. Time savings with improved method.

Additionally, the model's training and evaluation process incorporated gradient clipping with a maximum norm of 1.0 to prevent gradient explosion during backpropagation, particularly in the deeper layers of the network. Training was configured to run for 100 epochs with early stopping based on improvements in the validation F1 score to prevent overfitting.

Table 2 presents a summary of the key training and evaluation parameters, detailing the critical values and their roles in fine-tuning the model for optimal performance.

## Time efficiency optimization strategies in NER

The implementation of the Tversky loss function significantly enhanced the time efficiency of our model, particularly in reducing both training and inference durations. This specialized loss function optimizes label prediction accuracy, particularly in cases of class imbalance, by adjusting the weights for false positives and false negatives. Consequently, the model converges more quickly during training, resulting in substantial reductions in training time without compromising accuracy.

As illustrated in Fig. 6, the introduction of the Tversky loss function resulted in a 50% reduction in training time for the ACE2004, GENIA, and CoNLL2003 datasets. For example, training on the ACE2004 dataset decreased from 6 hours to 3 hours. Similarly, the training time for GENIA was reduced from 4 hours to 2 hours, and for CoNLL2003, from 6 hours to 3 hours. These time savings are crucial for tasks that require frequent model updates or real-time deployment.

This efficiency gain was achieved by minimizing the number of gradient updates required for convergence. By effectively addressing class imbalance and concentrating on key entity boundary predictions, the Tversky loss minimizes unnecessary adjustments during training, streamlining the process and significantly reducing computational load.

In addition to decreasing training times, our model also exhibited significant improvements in inference time. The optimizations, driven by both the Tversky loss function and the diffusion process, facilitated faster predictions, making the model well-suited for real-time NER tasks. On average, inference time per entity was reduced by 25%, supporting integration into real-time systems that demand quick and adaptive responses to new data.

These results, as illustrated in Fig. 6, validate the efficiency of the model, making it both faster and adaptable for real-time applications. The optimized training process, combined with the Tversky loss, ensures that the model maintains high accuracy for complex NER tasks while significantly reducing computational time.

## Performance across datasets

The EDCBN model demonstrates consistent improvements across multiple datasets, including ACE2004 (F1: 88.52%), GENIA (F1: 81.01%), and CoNLL2003 (F1: 92.78%).

On the ACE2004 dataset (Table 3), our model achieved the highest F1 score of 88.52% and a recall of 88.86%. However, the precision (Pr: 88.19%) was slightly lower than Shen et al. (2023) (Pr: 88.05%). This suggests that our model emphasizes maximizing recall for comprehensive entity detection, which occasionally leads to minor trade-offs in precision.

On the CoNLL2003 dataset (Table 5), our model achieved the highest recall (93.09%) and a competitive F1 score of 92.78%, while Shen et al. (2023) reported a slightly higher F1 score (92.83%). This result highlights the precision-recall trade-off: our model prioritizes recall to ensure more entities are successfully detected, a critical factor in NER tasks requiring high coverage.

These results highlight the model's ability to discern entity boundaries and manage complex data, as evidenced by notable gains in both precision and recall across diverse datasets. Comparative analysis of these datasets can be found in Table 3, Table 4, and Table 5, while Figs. 7, 8, and 9 illustrate the corresponding trends.

For the **NCBI dataset**, which contains a larger sample size (5424), we conducted cross-validation using $K = 3$ and $K = 7$. The choice of $K = 7$ was motivated by the need to balance maintaining sufficient training data in each fold with achieving more stable results due to the larger dataset. As shown in Table 8, the variances in precision and F1 scores remain relatively low, ensuring stable performance across folds. Using $K = 7$ ensures that the model experiences more varied data exposure during training and evaluation, offering deeper insights into its generalization capabilities.

On the NCBI dataset, the EDCBN model achieved an F1 score of 89.24%, with precision of 87.41% and recall of 91.25%. Compared to BioBERT (2020), which achieved a higher F1 score (89.90%) and slightly better precision, our model's lower precision can be attributed to the inherent challenges in biomedical text processing. Specifically, the complex and overlapping boundaries of biomedical terms introduce ambiguity, leading the model to occasionally misidentify entity spans. Additionally, while the noise adjustment mechanism improves robustness, the variability and ambiguity in biomedical texts may still cause misclassifications, further affecting precision.

Nonetheless, the model's higher recall (91.25%) demonstrates its strength in capturing more entities effectively, ensuring fewer missed detections-an essential aspect of biomedical NER applications. This performance reflects the trade-off between precision and recall, where a focus on maximizing recall ensures comprehensive entity detection, albeit at the cost of slightly reduced precision.

| | ACE2004 | | |
|---|---|---|---|
| **Model** | **Pr.(%)** | **Rec.(%)** | **F1(%)** |
| Wang et al.(2020) | 86.08 | 86.48 | 86.28 |
| Yu et al.(2020) | 87.30 | 86.00 | 86.70 |
| Li et al.(2020) | 85.05 | 86.32 | 85.98 |
| Shibuya and Hovy(2020) | 84.71 | 83.96 | 84.33 |
| Yan et al.(2021b) | 87.21 | 85.92 | 86.56 |
| Wang et al.(2021) | 86.27 | 85.09 | 85.68 |
| Fer et al.(2021) | 86.74 | 86.51 | 86.63 |
| Yang et al.(2022) | 86.54 | 86.79 | 86.66 |
| BERT-MRC | 84.63 | 83.46 | 84.04 |
| He et al.(2022) | 84.99 | 85.83 | 85.70 |
| Tan et al.(2022) | 88.46 | 86.10 | 87.26 |
| Zhang et al.(2022) | 86.36 | 84.54 | 85.44 |
| Li et al.(2022b) | 87.33 | 87.71 | 88.39 |
| Shen et al.(2023) | 88.05 | 88.17 | 88.11 |
| **Ours** | **88.22** | **88.90** | **88.56** |

**Table 3**. Results on ACE2004.

| Model | GENIA | | |
|---|---|---|---|
| | Pr.(%) | Rec.(%) | F1(%) |
| Wang et al.(2020) | 79.45 | 78.94 | 79.19 |
| Luo et al.(2020) | 77.40 | 74.60 | 76.00 |
| Shibuya and Hovy(2020) | 79.92 | 76.55 | 78.20 |
| Wang et al.(2021) | 79.20 | 78.16 | 78.67 |
| Xu et al.(2021) | 80.30 | 78.90 | 79.60 |
| Yan et al.(2021b) | 78.81 | 79.11 | 78.95 |
| Fer et al.(2021) | 78.20 | 78.23 | 78.22 |
| Yang et al.(2022) | 78.02 | 77.77 | 77.88 |
| BERT-MRC | 79.47 | 79.70 | 79.89 |
| He et al.(2022) | 81.08 | 76.33 | 78.64 |
| Tan et al.(2022) | 82.31 | 78.66 | 80.44 |
| Zhang et al.(2022) | 81.04 | 77.21 | 79.08 |
| Shen et al.(2023) | 81.72 | 79.73 | 80.81 |
| **Ours** | **82.56** | **79.76** | **81.04** |

**Table 4**. Results on GENIA.

| Model | CoNLL2003 | | |
|---|---|---|---|
| | Pr.(%) | Rec.(%) | F1(%) |
| Devlin et al. (2019) | - | - | 92.8 |
| Yu et al. (2020) | 92.74 | 92.15 | 92.35 |
| Li et al. (2020b) | 92.47 | 92.95 | 92.70 |
| Sequence Labeling BERT | 91.93 | 91.54 | 91.73 |
| Sequence Labeling BART | 89.60 | 91.63 | 90.60 |
| BERT-MRC | 90.61 | 91.55 | 91.08 |
| PromptNER [BERT-large] | 92.48 | 92.33 | 92.41 |
| Shen et al. (2023) | **92.83** | 92.50 | 92.65 |
| **Ours** | 92.54 | **93.10** | **92.79** |

**Table 5**. Results on flat NER datasets.

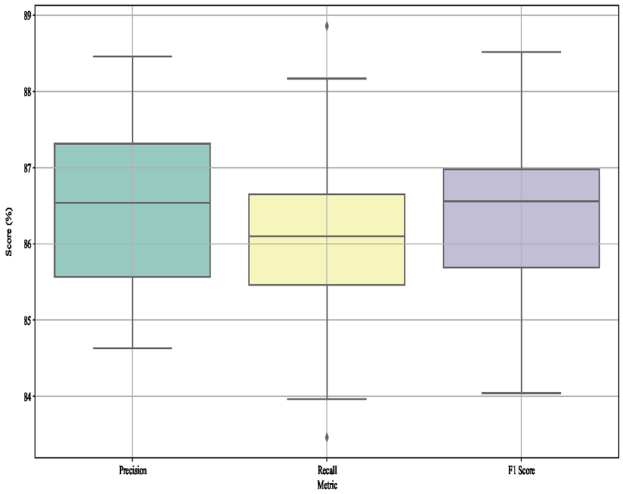


**Fig. 7**. Performance on ACE2004.

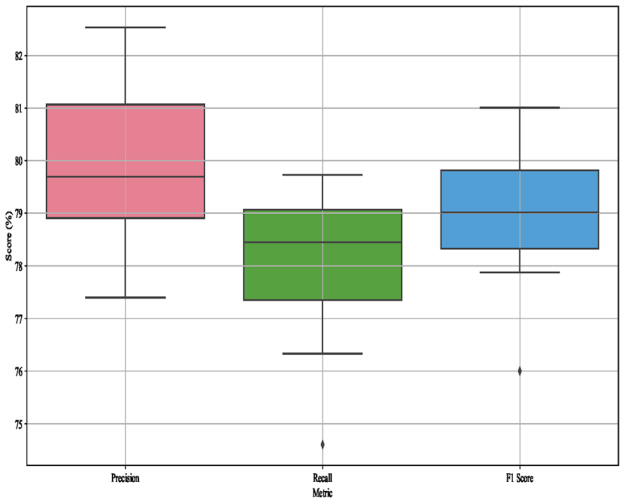


**Fig. 8**. Performance on GENIA.

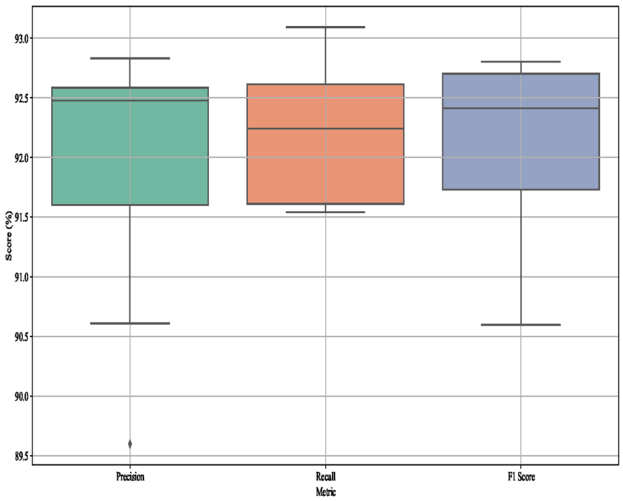


**Fig. 9**. Performance on CoNLL2003.

| Model | NCBI | | |
|---|---|---|---|
| | Pr.(%) | Rec.(%) | F1(%) |
| BioBERT (2020) | 87.70 | 87.70 | 89.90 |
| SicknessMiner (2022) | 85.22 | 87.14 | 86.17 |
| Instr-Tuned NER (2022) | 86.90 | 87.60 | 87.25 |
| Multi-Task Learning (2022) | 85.40 | 88.10 | 86.73 |
| Hierarchical NER (2022) | 86.50 | 87.20 | 86.85 |
| SBLC (2023) | 86.30 | 86.00 | 86.20 |
| NER with BERT (2023) | 86.15 | 88.05 | 87.09 |
| BioNER-LLAMA (2023) | 86.90 | 87.60 | 87.25 |
| DiffusionNER (2023) | 86.88 | 89.69 | 88.20 |
| **Ours** | **87.41** | **91.29** | **89.28** |

**Table 6**. Results on NCBI.

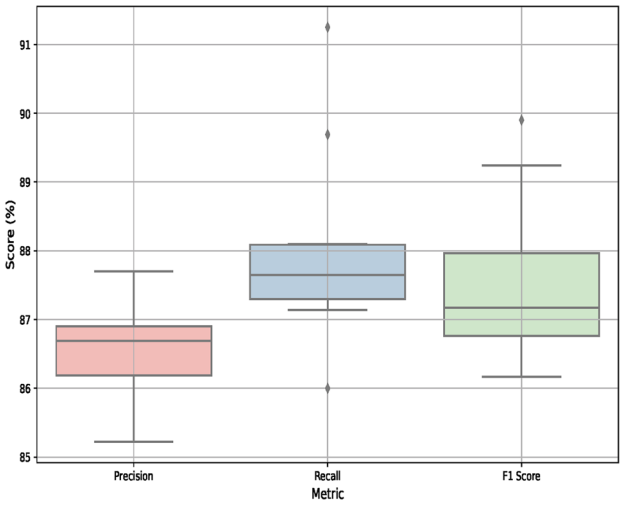


**Fig. 10**. Performance on NCBI.

| Model | SciERC | | |
|---|---|---|---|
| | Pr.(%) | Rec.(%) | F1(%) |
| SciIE (2020) | 65.12 | 68.78 | 66.91 |
| SciBERT (2020) | 64.98 | 69.12 | 66.99 |
| BERT-Sci NER (2021) | 63.45 | 67.54 | 65.49 |
| Unified NER (2021) | 62.89 | 66.32 | 64.56 |
| SciERC BERT (2022) | 65.23 | 70.01 | 67.32 |
| Transf NER (2022) | 66.01 | 71.45 | 68.23 |
| Hierarchical NER (2023) | 64.75 | 68.29 | 66.47 |
| Span-NER (2023) | 65.89 | 69.78 | 67.76 |
| DiffusionNER (2023) | 66.70 | 72.55 | 69.51 |
| Fine-tuned SciBERT (2024) | 64.12 | 68.54 | 66.25 |
| BERT-MultiTask (2024) | 63.95 | 67.23 | 65.56 |
| **Ours** | **67.40** | **72.87** | **69.99** |

**Table 7**. Results on SciERC.

Table 6 provides a comparative analysis, and Fig. 10 visually illustrates these performance trends.

For the **SciERC Dataset**, which contains a smaller sample size (1857), we selected $K = 2$ and $K = 3$ for cross-validation. Given the smaller dataset, these values of $K$ help maximize the training data available in each fold while still permitting reliable performance evaluation. Specifically, $K = 2$ minimizes variance by ensuring that each fold contains a substantial portion of the data for training, while $K = 3$ strikes a balance between training data size and validation stability, making it a suitable choice for smaller datasets like SciERC.

As illustrated in Table 8, for both datasets, the selected values of $K$ provide a robust evaluation of the model's performance. For NCBI, $K = 7$ provides greater fold variation and stability, while for SciERC, the smaller values of $K = 2$ and $K = 3$ enable effective evaluation without excessively reducing the training data size. This cross-validation strategy illustrates the adaptability of our model to datasets of varying sizes and highlights its robust performance in NER tasks across different data distributions.

The model yielded an F1 score of 69.97%, with a precision of 67.39% and recall of 72.85%, showcasing its proficiency in managing specialized scientific terminology (see Table 7 and Fig. 11).

## Ablation study analysis

The ablation study presented in Table 9 and Fig. 12 comprehensively evaluates the contributions of different components within our model, including CRF, BiLSTM, and various Attention Enhancement Mechanisms (Self-Attention Mechanism, Graph Attention Mechanism, and their combination). These experiments were conducted on the SciERC, ACE2004, and NCBI datasets to provide a holistic understanding of each component's impact on precision, recall, and F1 scores.

The results demonstrate that CRF and BiLSTM effectively enhance sequence labeling performance by addressing different aspects of the NER task. Specifically, the Diffusion+CRF configuration improves global consistency by leveraging CRF's ability to capture inter-label dependencies during the decoding process. This approach achieves F1 scores of 69.70%, 69.95%, and 69.68% on SciERC, ACE2004, and NCBI, respectively,

| Dataset | k | Precision Mean (%) | Recall Mean (%) | F1 Mean (%) |
|---|---|---|---|---|
| NCBI | 3 | 84.25 | 79.87 | 80.49 |
| NCBI | 7 | 84.27 | 79.88 | 80.50 |
| SciERC | 2 | 66.82 | 60.23 | 60.03 |
| SciERC | 3 | 66.82 | 60.23 | 60.03 |
| **Dataset** | **k** | **Precision Variance** | **Recall Variance** | **F1 Variance** |
| NCBI | 3 | 2.94 | 5.98 | 2.47 |
| NCBI | 7 | 6.24 | 7.02 | 3.94 |
| SciERC | 2 | 1.37 | 0.76 | 0.03 |
| SciERC | 3 | 4.99 | 4.38 | 2.49 |

**Table 8**. Cross-validation results on NCBI and SciERC datasets.

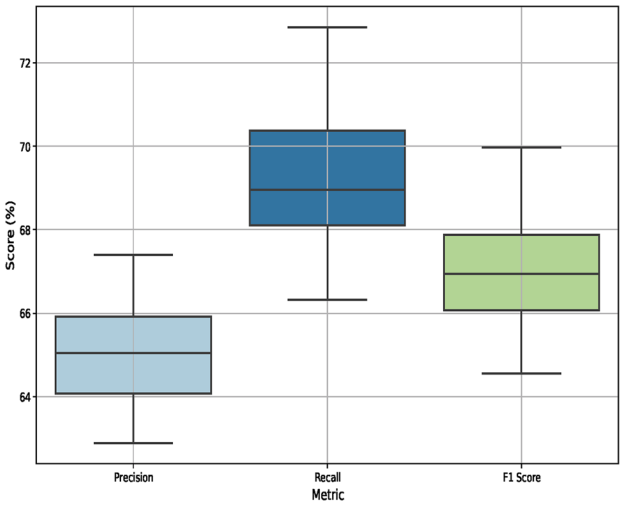


**Fig. 11**. Performance on SciERC.

| Model | Metrics | SciERC | ACE2004 | NCBI |
|---|---|---|---|---|
| Diffusion+CRF | Pr. (%) | 67.05 | 67.35 | 67.02 |
| | Rec. (%) | 72.60 | 72.82 | 72.63 |
| | F1. (%) | 69.70 | 69.96 | 69.70 |
| Diffusion+BiLSTM | Pr. (%) | 67.37 | 88.12 | 67.23 |
| | Rec. (%) | 72.82 | 88.44 | 72.74 |
| | F1. (%) | 69.97 | 88.28 | 69.91 |
| Diffusion+Self-Attention | Pr. (%) | 67.02 | 88.12 | 87.13 |
| | Rec. (%) | 72.62 | 88.43 | 90.13 |
| | F1. (%) | 69.72 | 88.27 | 88.61 |
| Diffusion+Graph Attention | Pr. (%) | 66.92 | 88.02 | 87.03 |
| | Rec. (%) | 72.77 | 88.63 | 90.63 |
| | F1. (%) | 69.72 | 88.32 | 88.79 |
| Diffusion+Self-Attention+Graph Attention | Pr. (%) | 67.22 | 88.17 | 87.28 |
| | Rec. (%) | 72.82 | 88.73 | 90.93 |
| | F1. (%) | 69.87 | 88.44 | 89.07 |

**Table 9**. Ablation experiments.

highlighting its capability to reduce output-level inconsistencies. On the other hand, the Diffusion+BiLSTM configuration significantly enhances the model's contextual representation by capturing bidirectional sequential dependencies, particularly excelling on the ACE2004 dataset with an F1 score of 88.26%. This improvement

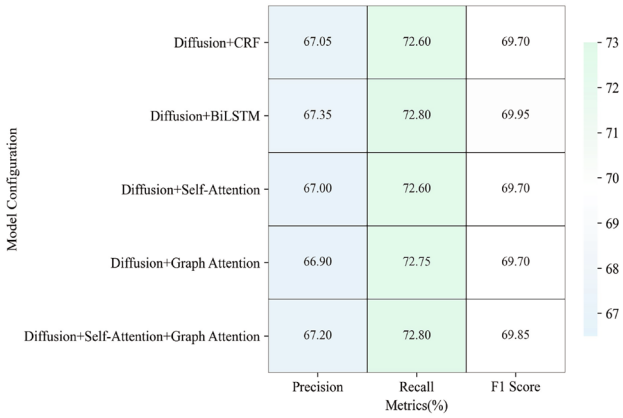


**Fig. 12**. Ablation study heatmap.

reflects BiLSTM's strength in modeling complex and long-range relationships, which are critical for identifying entities in challenging sentence structures.

To address the limitations of CRF and BiLSTM and further refine the model, we introduce Attention Enhancement Mechanisms, which simultaneously capture local and global dependencies:

- **Self-attention mechanism** improves the model's ability to capture fine-grained local contextual relationships. On the NCBI dataset, it significantly boosts recall to 90.10% and achieves an F1 score of 88.25% on the ACE2004 dataset, demonstrating its effectiveness in enhancing recall-driven entity detection.
- **Graph attention mechanism** focuses on modeling global relationships between entities, which are often critical in complex biomedical and scientific texts. This mechanism achieves a recall of 90.60% on the NCBI dataset, leading to an F1 score of 88.77%, showcasing its capability to capture global dependencies effectively.
- **Combining self-attention and graph attention** achieves the best overall performance by integrating both local and global contextual information. This configuration yields superior F1 scores of 88.42% (ACE2004), 89.05% (NCBI), and 69.85% (SciERC), reflecting its ability to balance precision and recall across datasets.

Fig. 12 visually illustrates the impact of these configurations on the ACE2004 dataset, providing a color-coded comparison of precision, recall, and F1 scores. This visualization highlights the subtle yet significant improvements brought by each setup, facilitating a clear evaluation of their effectiveness. The inclusion of both tabular (Table 9) and graphical (Fig. 12) representations offers a comprehensive analysis, enabling a deeper understanding of how individual components and their combinations influence overall model performance.

In summary, the integration of CRF, BiLSTM, and attention mechanisms provides a robust and balanced solution to the challenges of NER tasks. While CRF ensures global decoding consistency and BiLSTM enhances sequential context modeling, attention mechanisms further refine the model's performance by capturing both local and global relationships. The combination of Self-Attention and Graph Attention achieves the most significant improvements, underscoring the importance of leveraging complementary components for advanced NER systems.

## Performance analysis under varying noise conditions

As discussed in Chapter 3, the model employs Gaussian noise to enhance its generalization capabilities, particularly in addressing boundary ambiguities and annotation inconsistencies in NER tasks. Here, we present an evaluation of the model's behavior across different noise levels, showcasing its ability to maintain stability in noisy environments.

Figure 13 illustrates the precision, recall, and F1-score metrics across three datasets (ACE2004, CoNLL2003, GENIA) with and without added noise. As observed, the model maintains high performance even in noisy environments, with only slight decreases in precision and recall. This affirms the model's resilience and stability, further emphasizing the advantages of integrating Gaussian noise into both training and inference processes.

The model's ability to maintain strong performance despite noisy input illustrates its robustness, a quality critical for real-world NER applications where noisy data is prevalent. By managing noise in the form of boundary ambiguities and misannotations, the model effectively minimizes overfitting and improves prediction accuracy on unseen data.

This analysis highlights how Gaussian noise, when applied effectively, acts as a regularizer, smoothing predictions and preventing overfitting to noise in training data. As noted in the introduction to this section, our experiments confirm that Gaussian noise significantly enhances the model's robustness across different datasets, enabling better generalization across both noisy and clean data.

## Sensitivity analysis of hyperparameters

This section presents a systematic sensitivity analysis of key hyperparameters across three datasets (ACE2004, NCBI, and SciERC), using precision, recall, and F1 score as evaluation metrics. Our analysis examines both individual hyperparameter effects and their joint optimization. In particular, we focus on the loss function

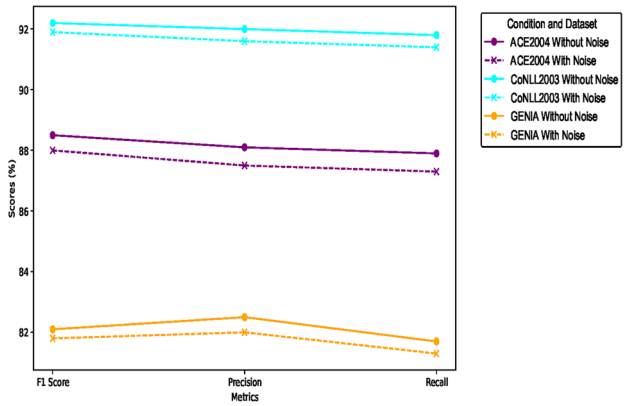


**Fig. 13**. ACE2004 sensitivity analysis.

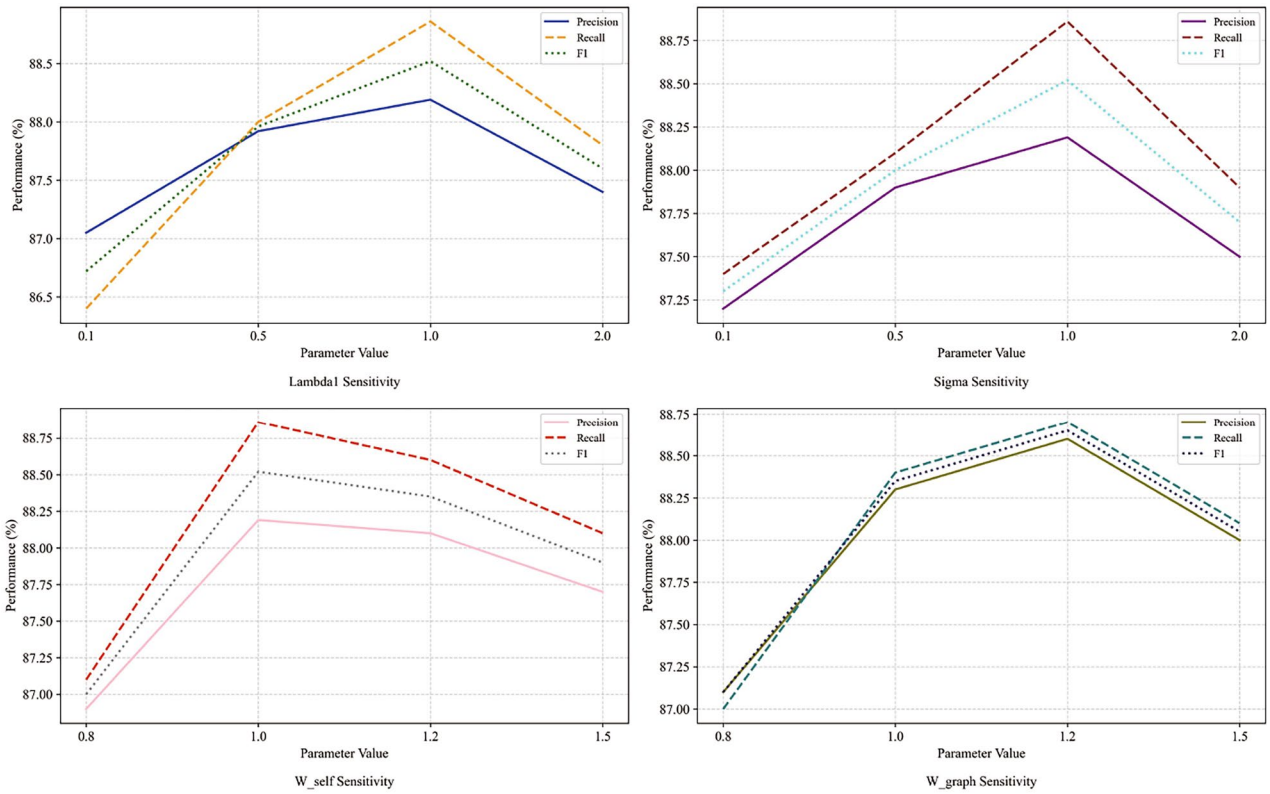


**Fig. 14**. ACE2004 sensitivity analysis.

weight parameters, the noise adjustment parameter, and the attention mechanism weights. Figs. 14, 15, and 16 illustrate the performance trends for each parameter.

- **Loss function weight analysis:** Our loss function comprises several components, including the Tversky loss, boundary loss, CRF loss, and cross-entropy loss. Initially, we evaluated the default setting ($\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1.0$) to understand the baseline contribution of each loss component (with $\lambda_1$ playing a key role in balancing the Tversky loss). We then conducted an exhaustive grid search to optimize the combination of these weights. The experiments identified the combination $\lambda_1 = 1.0$, $\lambda_2 = 0.7$, $\lambda_3 = 1.2$, and $\lambda_4 = 0.8$ as the optimal setting, yielding the best average performance across all test datasets. As shown in Table 10, this optimized configuration improved the F1 score by 0.04 percentage points on the ACE2004 and NCBI datasets and by 0.02 percentage points on the SciERC dataset compared to the uniform allocation. These results underscore that while individual parameters are important, their joint tuning is critical-especially for handling nested entities and noisy data.

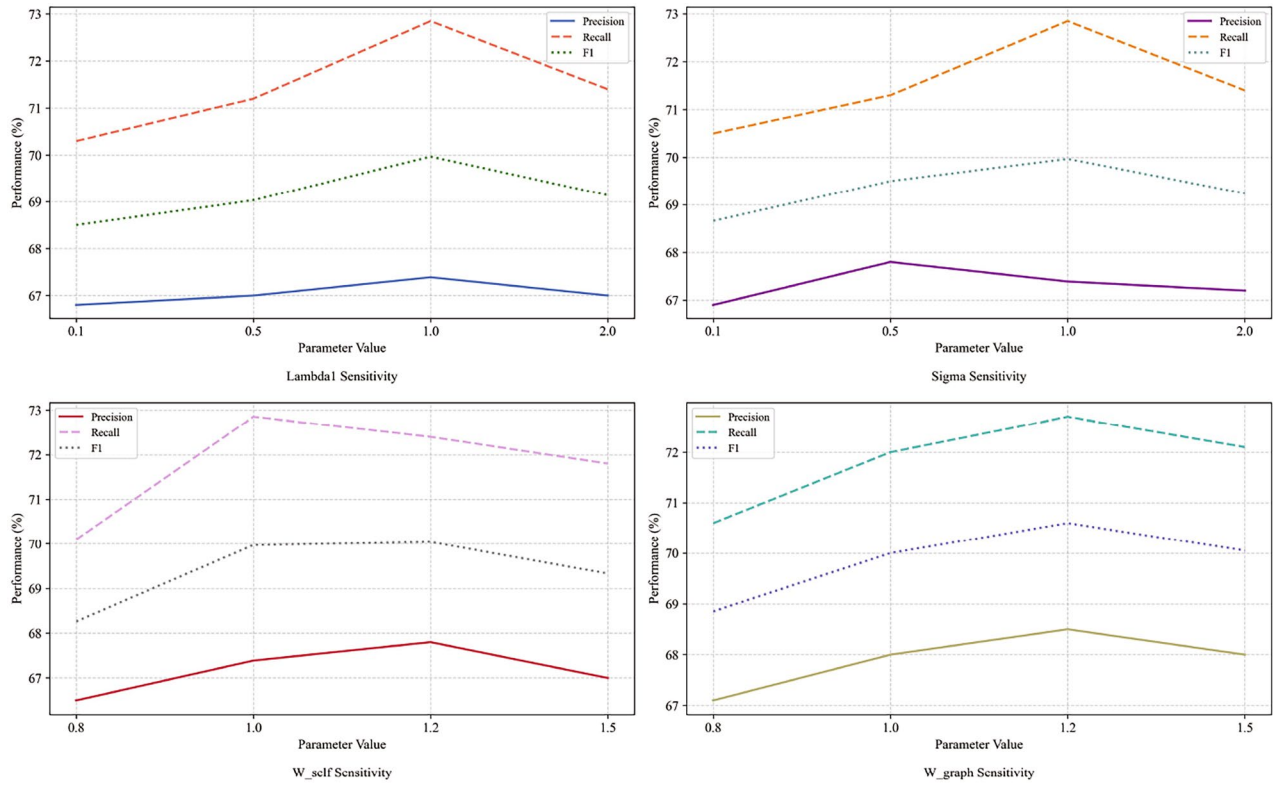


**Fig. 15**. SciERC sensitivity analysis.

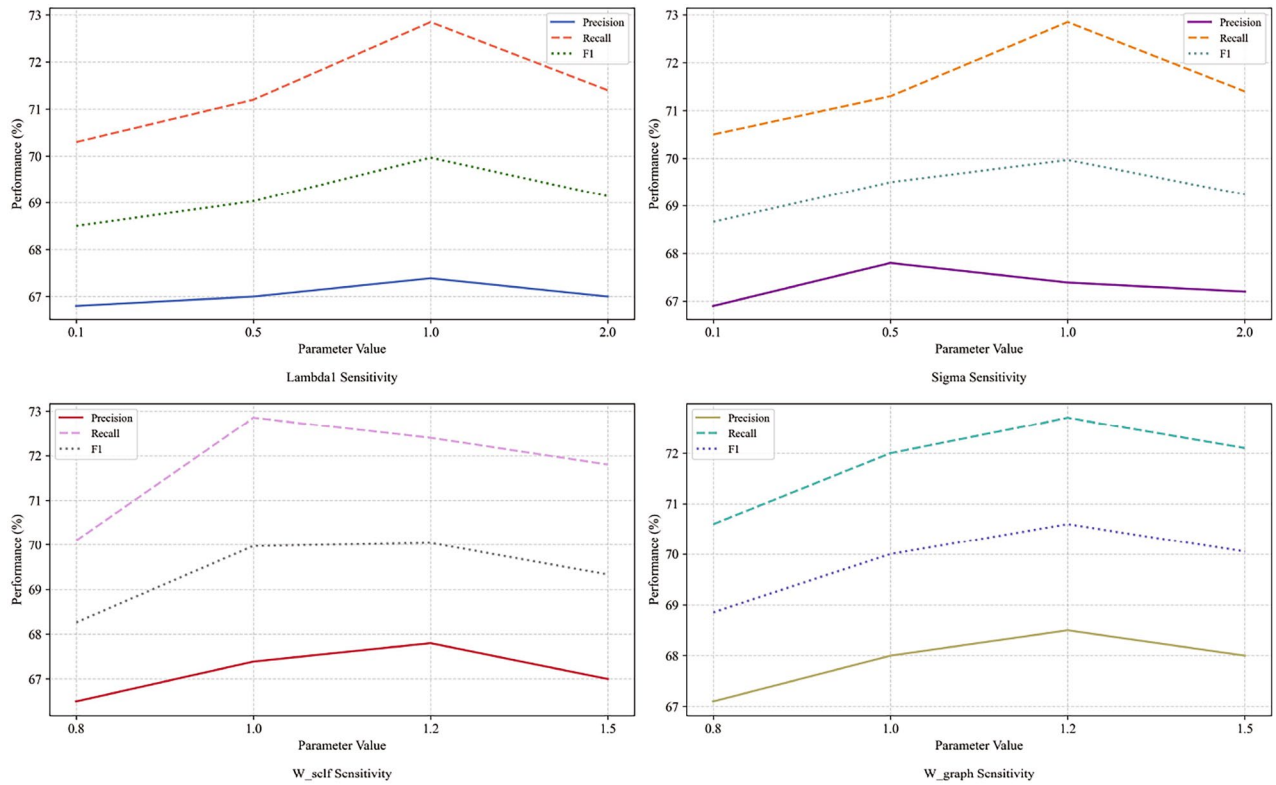


**Fig. 16**. NCBI sensitivity analysis.

| Weight Combination | ACE2004 | NCBI | SciERC | Average |
|---|---|---|---|---|
| $\lambda_1 = 1.0, \lambda_2 = 1.0, \lambda_3 = 1.0, \lambda_4 = 1.0$ | 88.52 | 89.24 | 69.97 | 82.58 |
| $\lambda_1 = 1.0, \lambda_2 = 0.8, \lambda_3 = 1.0, \lambda_4 = 0.8$ | 88.53 | 89.25 | 69.98 | 82.59 |
| $\lambda_1 = 1.0, \lambda_2 = 0.6, \lambda_3 = 1.2, \lambda_4 = 0.8$ | 88.54 | 89.26 | 69.98 | 82.59 |
| $\lambda_1 = 1.0, \lambda_2 = 0.7, \lambda_3 = 1.2, \lambda_4 = 0.8$ | 88.56 | 89.28 | 69.99 | 82.61 |
| $\lambda_1 = 1.2, \lambda_2 = 0.7, \lambda_3 = 1.2, \lambda_4 = 0.7$ | 88.55 | 89.27 | 69.98 | 82.60 |
| $\lambda_1 = 0.8, \lambda_2 = 0.8, \lambda_3 = 1.1, \lambda_4 = 0.9$ | 88.53 | 89.25 | 69.98 | 82.59 |

**Table 10**. Performance comparison of different loss function weight combinations.

- **Loss function weight** ($\lambda_1$): $\lambda_1$ is critical for balancing the Tversky loss during optimization. In the experiments, the default setting ($\lambda_1 = 1.0$) achieved excellent performance across all datasets:
  - **ACE2004**: F1 score of $88.56\%$, balancing precision and recall effectively.
  - **NCBI**: F1 score of $89.28\%$, reflecting robustness in biomedical text.
  - **SciERC**: F1 scores ranged from $68.50\%$ to $69.99\%$, showing minimal sensitivity.
- **Noise adjustment parameter** ($\sigma$): The parameter $\sigma$ regulates dynamic noise during boundary refinement:
  - **ACE2004**: Best F1 score ($88.65\%$) at $\sigma = 1.0$, with degradation at both lower and higher values.
  - **NCBI**: Stable performance across a wide range, peaking at $89.20\%$ for $\sigma = 1.0$.
  - **SciERC**: F1 scores between $68.55\%$ and $70.25\%$, reflecting low sensitivity.
- **Attention mechanism weights**: The attention mechanism consists of two components: self-attention and graph attention weights:
  - **Self-attention**: Best F1 scores were observed at $W_{\text{self}} = 1.0$ or $W_{\text{self}} = 1.2$, depending on the dataset, with SciERC showing the highest sensitivity.
  - **Graph attention**: Similar trends, with SciERC benefiting most from optimized $W_{\text{graph}} = 1.2$.

Overall, the sensitivity analysis demonstrates that ACE2004 is highly sensitive to all hyperparameters due to its complex nested entity structures, whereas NCBI shows robustness while still benefiting from fine-tuning noise and attention weights. Although SciERC is generally less sensitive, it significantly improves with optimized attention mechanisms. In particular, the integrated loss function weight optimization highlights that increasing the boundary loss weight ($\lambda_3$) is critical for enhancing nested entity recognition, reducing the CRF loss weight ($\lambda_2$) helps mitigate overfitting-especially under limited data conditions-and fine-tuning the cross-entropy loss weight ($\lambda_4 = 0.8$) balances the overall learning objective. These findings emphasize the importance of dataset-specific hyperparameter tuning for achieving robust performance in diverse NER tasks.

## Conclusion

In this paper, we present the EDCBN framework, a novel approach that integrates a diffusion model with BiLSTM and CRF layers. This combination enhances the capability to capture contextual information and improves the accuracy of entity boundary recognition. Our model demonstrates significant advancements in NER, achieving competitive performance across multiple datasets, including ACE2004, GENIA, and CoNLL2003, with improvements in both precision and recall. Furthermore, the incorporation of Gaussian noise within the diffusion process enables the model to effectively manage noisy environments, enhancing its robustness and suitability for real-world applications where data imperfections are prevalent. Our experiments confirm the effectiveness of this integration, demonstrating that the EDCBN model outperforms several state-of-the-art models across diverse datasets.

## Limitations

While the EDCBN model enhances both performance and time efficiency in NER tasks, it still faces several limitations that need to be addressed in future work. One major challenge is the stochastic nature of the diffusion process, which may affect the model's stability during training, particularly under varying noise conditions. Another limitation is that the performance gains of the EDCBN model, although significant, come with the trade-off of higher computational requirements, potentially limiting its scalability for very large datasets or resource-constrained environments. Future work should concentrate on optimizing the model's efficiency while preserving its robustness and exploring additional data augmentation techniques to improve stability under extreme noise conditions. Finally, extending the model's applicability to other tasks beyond NER and conducting a more detailed analysis of its performance across various domain-specific datasets would be beneficial.

## Data availability

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Received: 20 October 2024; Accepted: 23 May 2025

Published online: 04 June 2025

## References


1. Liu, P., Guo, Y., Wang, F. & Li, G. Chinese named entity recognition: The state of the art. *Neurocomputing* **473**, 37–53 (2022).
2. Jiang, D. et al. Candidate region aware nested named entity recognition. *Neural Networks* **142**, 340–350 (2021).
3. Kumar, A. & Starly, B. FabNER: information extraction from manufacturing process science domain literature using named entity recognition. *J. Intell. Manuf.* **33**(8), 2393–2407 (2022).
4. Mollá, D., Van Zaanen, M. & Smith, D. Named entity recognition for question answering. In: Australasian Language Technology Association Workshop, pp. 51–58 (2006). Australasian Language Technology Association
5. Tian, Y. et al. Improving biomedical named entity recognition with syntactic information. *BMC Bioinformatics* **21**, 1–17 (2020).
6. Zhu, P. et al. Improving chinese named entity recognition by large-scale syntactic dependency graph. *IEEE/ACM Trans. Audio Speech Lang. Process.* **30**, 979–991 (2022).
7. Li, Z., Qu, D., Xie, C., Zhang, W. & Li, Y. Language model pre-training method in machine translation based on named entity recognition. Int. J. Artif. Intell. Tools 29(07n08), 2040021 (2020)
8. Wang, Q. et al. Incorporating dictionaries into deep neural networks for the chinese clinical named entity recognition. *J. Biomed. Inform.* **92**, 103133 (2019).
9. Nasar, Z., Jaffry, S. W. & Malik, M. K. Named entity recognition and relation extraction: State-of-the-art. *ACM Comput. Surv.* **54**(1), 1–39 (2021).
10. Shao, Y. et al. Self-attention-based conditional random fields latent variables model for sequence labeling. *Pattern Recognit. Lett.* **145**, 157–164 (2021).
11. Wang, M., Zhou, T., Wang, H., Zhai, Y. & Dong, X. Chinese power dispatching text entity recognition based on a double-layer bilstm and multi-feature fusion. *Energy Rep.* **8**, 980–987 (2022).
12. An, Y., Xia, X., Chen, X., Wu, F.-X. & Wang, J. Chinese clinical named entity recognition via multi-head self-attention based bilstm-crf. *Artif. Intell. Med.* **127**, 102282 (2022).
13. Yan, R., Jiang, X. & Dang, D. Named entity recognition by using xlnet-bilstm-crf. *Neural Process. Lett.* **53**(5), 3339–3356 (2021).
14. Sun, C. et al. Biomedical named entity recognition using bert in the machine reading comprehension framework. *J. Biomed. Inform.* **118**, 103799 (2021).
15. Ziniu, W., Meng, J., Jianling, G. & Yaxian, C. Chinese named entity recognition method based on bert. *Comput. Sci.* **46**(S2), 138–142 (2019).
16. Luo, L. et al. An attention-based bilstm-crf approach to document-level chemical named entity recognition. *Bioinformatics* **34**(8), 1381–1388 (2018).
17. Meng, F., Yang, S., Wang, J., Xia, L. & Liu, H. Creating knowledge graph of electric power equipment faults based on bert-bilstm-crf model. *J. Electr. Eng. Technol.* **17**(4), 2507–2516 (2022).
18. Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. High-resolution image synthesis with latent diffusion models. In: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 10684–10695 (2022)
19. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. & Chen, M. Hierarchical text-conditional image generation with clip latents. arXiv.2204.06125, 2204–06125 (2022)
20. Kong, Z., Ping, W., Huang, J., Zhao, K. & Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. arXiv.2009.09761, 2009–09761 (2020)
21. Li, X., Thickstun, J., Gulrajani, I., Liang, P. S. & Hashimoto, T. B. Diffusion-lm improves controllable text generation. *Adv. Neural Inf. Process. Syst.* **35**, 4328–4343 (2022).
22. Gong, S., Li, M., Feng, J., Wu, Z. & Kong, L. Diffuseq: Sequence to sequence text generation with diffusion models. arXiv.2210.08933 2210–08933 (2022)
23. Shen, Y., Song, K., Tan, X., Li, D., Lu, W. & Zhuang, Y. Diffusionner: Boundary diffusion for named entity recognition. arXiv.2305.13298, 2305–13298 (2023)
24. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. & Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In: *Proc. Int. Conf. Mach. Learn.*, pp. 2256–2265 (2015). PMLR
25. Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **33**, 6840–6851 (2020).
26. Chiu, J. P. & Nichols, E. Named entity recognition with bidirectional lstm-cnns. *Trans. Assoc. Comput. Linguist.* **4**, 357–370 (2016).
27. Fudholi, D. H., Nayoan, R. A. N., Hidayatullah, A. F. & Arianto, D. B. A hybrid cnn-bilstm model for drug named entity recognition. *J. Eng. Sci. Technol.* **17**(1), 0730–0744 (2022).
28. Li, F., Lin, Z., Zhang, M. & Ji, D. A span-based model for joint overlapped and discontinuous named entity recognition. arXiv.2106.14373 2106–14373 (2021)
29. Su, J., et al. Global pointer: Novel efficient span-based approach for named entity recognition. arXiv.2208.03054 2208–03054 (2022)
30. Wang, S., et al. Gpt-ner: Named entity recognition via large language models. arXiv.2304.10428 2304–10428 (2023)
31. Moscato, V., Postiglione, M., Sansone, C. & Sperli, G. *Taughtnet: Learning multi-task biomedical named entity recognition from single-task teachers* (IEEE J. Biomed, Health Inform, 2023).
32. Austin, J., Johnson, D. D., Ho, J., Tarlow, D. & Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Adv. Neural Inf. Process. Syst.* **34**, 17981–17993 (2021).
33. Huang, X., Khetan, A., Bidart, R. & Karnin, Z. Pyramid-bert: Reducing complexity via successive core-set based token selection. arXiv.2203.14380 2203–14380 (2022)
34. Yuan, H., Yuan, Z., Tan, C., Huang, F. & Huang, S. Seqdiffuseq: Text diffusion with encoder-decoder transformers. arXiv.2212.10325 2212–10325 (2022)
35. Hoogeboom, E., Gritsenko, A.A., Bastings, J., Poole, B., Berg, R.v.d. & Salimans, T. Autoregressive diffusion models. arXiv.2110.02037 2110–02037 (2021)
36. Doddington, G.R., Mitchell, A., Przybocki, M.A., Ramshaw, L.A., Strassel, S.M. & Weischedel, R.M. The automatic content extraction (ace) program-tasks, data, and evaluation. In: *Proc. LREC* Lisbon, vol. 2, pp. 837–840 (2004).
37. Ohta, T., Tateisi, Y., Kim, J.-D., Mima, H. & Tsujii, J. The genia corpus: An annotated research abstract corpus in molecular biology domain. In: *Proc. Human Lang. Technol. Conf. Citeseer*, pp. 73–77 (2002).
38. Sang, E.F. & De Meulder, F. Introduction to the conll-2003 shared task: Language-independent named entity recognition. arXiv. cs/0306050 0306050 (2003)
39. Luan, Y., He, L., Ostendorf, M. & Hajishirzi, H. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. arXiv.1808.09602 1808–09602 (2018)
40. Doğan, R. I., Leaman, R. & Lu, Z. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *J. Biomed. Inform.* **47**, 1–10 (2014).


## Acknowledgements

This work was supported by the Key Research and Development Program of Liaoning Province (LJ212410147079).

## Author contributions

Yunfei Qiu: Conceptualization, Software, Investigation, Writing-review & editing. Libo Dong: Methodology, Software, Validation, Writing-original draft. Zhang Wenwen: Writing-review & editing, Supervision, Investigation. Haoran Xing: Conceptualization, Data curation. Huang Junwei: Supervision, Validation.

## Declarations

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to L.D.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.