

LoCoDec: Robust Decoding for LLM-based Sequence-Structured Prediction

Chang Liu¹, Peiyan Wang^{1*}, Yubo Feng², Libang Wang¹, Feiyan Jiang¹, Yan Song³

¹Shenyang Aerospace University, ²Dalian University of Technology,

³University of Science and Technology of China

{liuchang82, wanglibang, jiangfeiyan}@stu.sau.edu.cn,

wangpy@sau.edu.cn, argmax@126.com, clkson@gmail.com

Abstract

Many Natural Language Processing (NLP) tasks, including information extraction, dialogue understanding, and sentiment analysis, can be formulated as structured prediction over sequences or spans. Although Large Language Model (LLM)-based methods have achieved strong performance on standard benchmarks for such tasks, their robustness under natural sentence-level perturbations and their controllability in producing valid structures remain insufficiently studied. In this paper, we propose **Locality-aware Constrained Decoding (LoCoDec)**, a method for LLM-based structured prediction over sequences and spans that formulates decoding as constrained token paraphrasing. LoCoDec combines neighborhood context, which reduces long-text noise, with structural decoding constraints that enforce source-faithful and well-formed outputs. Experiments on sixteen benchmarks of six tasks show that LoCoDec remains competitive in S-F1 and P-F1, while achieving the best R-F1 on all datasets. Ablation results show that removing neighborhood context reduces R-F1 by up to 16.23 points. Overall, LoCoDec improves the robustness and structural consistency of LLM-based structured prediction.

1 Introduction

Many Natural Language Processing (NLP) problems can be formulated as sequence-structured prediction tasks, such as Named Entity Recognition (NER), Part-of-Speech (POS) tagging, and Semantic Role Labeling (SRL). In this work, we use sequence-structured prediction to refer to tasks whose outputs can be serialized as token, span, or nested-span labels aligned with the source sequence. Recent work has increasingly formulated such tasks as conditional language modeling, allowing Large Language Models (LLMs) to serve as flexible predictors for source-aligned structured

*Corresponding author.

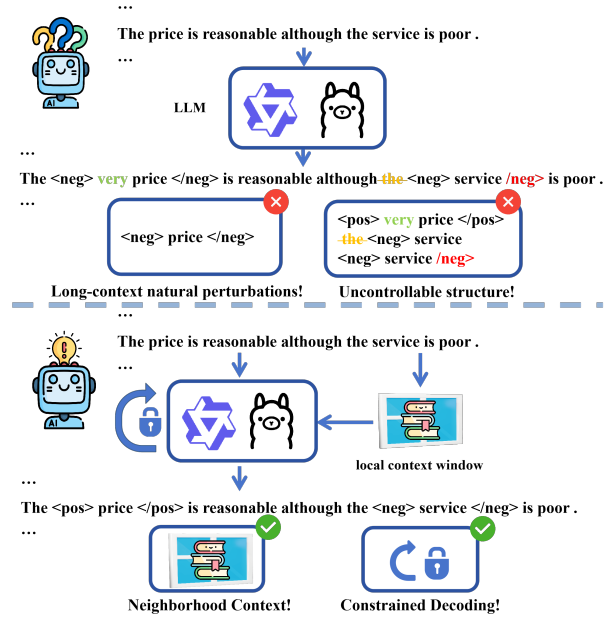


Figure 1: Challenges faced by LLM-based sequence-structured prediction in long-context scenarios (top) and our proposed LoCoDec method (bottom). “<label>” and “</label>” mark boundaries of Aspect-Based Sentiment Analysis (ABSA) task labels.

outputs (Alves et al., 2023; Kamalloo et al., 2023; Kim et al., 2024; Yang et al., 2024; Zhang et al., 2023; Wang et al., 2025).

Despite their strong performance, LLM-based sequence-structured predictors still face two major challenges. First, they are sensitive to input perturbations, where small changes in the surrounding context may lead to substantially different predictions (Chen et al., 2024b; Wang et al., 2023; Srinivasan and Vajjala, 2023). Second, their autoregressive generation process can produce invalid or unfaithful structures, which violate the strict output constraints required by sequence-structured prediction tasks. These issues become more severe in long-context settings, where irrelevant surrounding sentences may interfere with local boundary

and label decisions. Although prior work has explored constrained generation and grammar-guided decoding for LLMs (Le et al., 2024; Raspanti et al., 2025; Geng et al., 2024; Dong et al., 2025), most methods mainly constrain the output format. They do not fully address source-text faithfulness and robustness to long-context natural perturbations at the same time.

Robustness evaluation in NLP has focused on adversarial or synthetic perturbations (Das and Paik, 2022; Lin et al., 2021; Namysl et al., 2020). However, these perturbations are often linguistically unnatural and may not reflect real deployment scenarios. As noted by Wang et al. (2022), robustness conclusions drawn from artificial perturbations may have limited practical validity. In contrast, sentence-level natural perturbations, such as sentence concatenation and contextual expansion, frequently occur when sequence-structured prediction systems are applied to long documents or noisy contexts. Their impact on LLM-based sequence-structured prediction remains underexplored.

Motivated by this gap, we study LLM-based sequence-structured prediction under sentence-level non-adversarial perturbations in long-context settings. Following the fine-grained robustness categories of Drenkow et al. (2022), we focus on natural perturbations that preserve semantic plausibility while changing the contextual structure. We consider both Chinese and English sequence-structured prediction tasks and aim to answer the following research questions:

Q1: How can an LLM-based generation framework uniformly model heterogeneous sequence-structured prediction tasks?

Q2: How sensitive are LLM-based sequence-structured predictors to sentence-level natural perturbations under long-context settings?

Q3: How can robustness to such perturbations be improved while preserving structural correctness?

To address these questions, we formulate the considered tasks as *constrained token paraphrase tasks*. Instead of generating independent key-value pairs or span offsets, the model regenerates the input sequence and inserts labels into the copied text. This design enforces source-text faithfulness while still leveraging autoregressive generation. As shown in Fig. 1, we propose **Locality-aware Constrained Decoding (LoCoDec)**, which combines neighborhood context with structural decoding constraints. We summarize our findings as follows:

- Constrained token paraphrasing better preserves

source-text faithfulness than span-offset prediction in LLM-based sequence-structured prediction.

- Strong performance on unperturbed inputs does not necessarily translate into perturbation robustness, as LLM-based sequence-structured predictors remain sensitive to sentence-level natural perturbations.
- LoCoDec improves prediction consistency under sentence-level natural perturbations while maintaining well-formed source-aligned outputs and competitive standard accuracy.

2 Related Work

LLMs for Sequence-Structured Prediction.

LLMs have generation and reasoning abilities, but adapting autoregressive generation to sequence-structured prediction remains challenging (Chen et al., 2024a). BIO labeling is effective for discriminative sequence labeling, but it is less compatible with generative LLMs because subword tokenization can break token-label alignment and nested structures require more complex labels. Recent work has therefore reformulated source-aligned structured prediction as generation. CodeIE (Guo et al., 2024) and tabular prompting (Li et al., 2024) impose structured output formats, while GPT-NER (Wang et al., 2025) and multi-stage syntactic parsing (Tian et al., 2024) reduce generation errors through self-verification, external knowledge, or filtering. These methods mainly follow Structured Content Extraction (SCE) (Han et al., 2024) or Structured Span Extraction (SSE) (Ding et al., 2025). SCE lacks explicit surface-level alignment with the source text, while SSE relies on exact character offsets and is sensitive to boundary errors in long contexts. LoCoDec instead formulates the considered sequence-structured prediction tasks as *constrained token paraphrasing*, which better matches the autoregressive generation process of LLMs while keeping the output aligned with the input text.

Robustness under Sentence-Level Perturbations.

Robustness measures whether a model maintains expected outputs under distribution shift. Prior work distinguishes adversarial robustness from non-adversarial, or natural, robustness (Drenkow et al., 2022). Existing studies mainly examine adversarial settings, including malicious prompts (Zhu et al., 2024; Wang et al., 2023) and artificial perturbations such as character flips, word insertion, or

word-order changes (Namysl et al., 2020; Lin et al., 2021; Das and Paik, 2022; Srinivasan and Vajjala, 2023). However, results from synthetic perturbations may not transfer well to natural linguistic variations (Wang et al., 2022). This issue matters for long-text sequence-structured prediction, where distracting context can affect local boundary and label decisions.

The most closely related work is Liu et al. (2026), which studies sentence-level non-adversarial robustness for Chinese NER. Their work uses a concatenation-based perturbation method and an F1-based robustness metric. We adopt this robustness setting as an evaluation protocol, but LoCoDec differs from Liu et al. (2026) in both task scope and decoding mechanism. Unlike Liu et al. (2026), which focuses on flat Chinese NER, LoCoDec targets broader source-aligned sequence and span tasks. Furthermore, it ensures text faithfulness and structural validity through monotonic text alignment, stack-based constraints, boundary verification, tokenization repair, and KV-cache alignment. Therefore, LoCoDec should be regarded as a general, source-faithful constrained decoding method, rather than a direct reuse of an adaptive decoding strategy specialized for NER. A more detailed controlled comparison with Liu et al. (2026), including a fixed-window LoCoDec variant, is provided in Appendix E.

3 Method

In this section, we introduce LoCoDec, a locality-aware constrained decoding method for LLM-based sequence-structured prediction. LoCoDec is designed for tasks whose outputs can be represented as source-aligned token labels, spans, or nested spans. LoCoDec consists of two components: neighborhood context and constrained token paraphrasing. The workflow in Fig. 2 proceeds from input and prompt construction, to neighborhood context selection and KV-cache alignment, then to LLM continuation proposal, constrained control, and source-faithful output assembly. The method section follows the same logic: Section 3.1 formalizes the task, Section 3.2 describes neighborhood context selection and cache alignment, and Section 3.3 describes the constrained controller for text alignment, label validity, boundary validation, and tokenization repair. Further implementation details of the segment-level validation controller are provided in Appendix D.

3.1 Task Formulation

We formalize the sequence-structured prediction tasks with LLMs as a constrained token paraphrasing task, in which the LLM regenerates the input sequence while inserting task-specific labels at appropriate positions. This formulation applies to token-level labels, flat spans, and nested spans that can be serialized over the source sequence.

Given an input sequence $x = (x_1, x_2, \dots, x_n)$ and a task-specific semantic label set L (e.g., PER and ORG for NER), LoCoDec performs iterative constrained decoding using paired start and end labels derived from L . The decoding process is formulated as:

$$y = \bigoplus_{i=1}^m \mathcal{M}^{(i)}(P(\tau, x, L, x_{w,i}) \oplus y_{w,i}), \quad (1)$$

where $\mathcal{M}^{(i)}$ denotes the i -th invocation of the LLM, and $P(\tau, x, L, x_{w,i})$ is the prompt that contains the task description τ , the full input sequence x and the label set L . $x_{w,i}$ and $y_{w,i}$ are respectively the input neighborhood window and the output neighborhood window of the i -th iteration.

Each generated token y_i is either copied from the input sequence or selected from structural label set $\mathcal{T}(L)$: $y_i \in \{x_1, \dots, x_n\} \cup \mathcal{T}(L)$, where $\mathcal{T}(L) = \{\langle \ell \rangle, \langle / \ell \rangle \mid \ell \in L\}$. This means that the output sequence should preserve the original token order of x and insert task labels at valid positions.

3.2 Neighborhood Context

Input Neighborhood Window. Sequence-structured prediction often relies on local evidence around the corresponding source token or span, while excessive long-range context may introduce irrelevant information. To reduce such interference, we instantiate the local context $x_{w,i}$ in $P(\tau, x, L, x_{w,i})$ with a dynamic window.

Given an input sequence x , we identify dynamic boundary positions:

$$\mathcal{B} = \{b_1, b_2, \dots, b_K\}, \quad (2)$$

where each b_k denotes the right boundary of a dynamic segment. At decoding step i , let p be the current source-text pointer. We select a window $[a, c)$ that contains p and define $x_{w,i} = x_{a:c}$.

Since dynamic windows may fragment cross-sentence structures, we further expand the window according to the stack of start labels:

$$\mathcal{S} = [(\ell^{(1)}, o^{(1)}, s^{(1)}), \dots, (\ell^{(r)}, o^{(r)}, s^{(r)})], \quad (3)$$

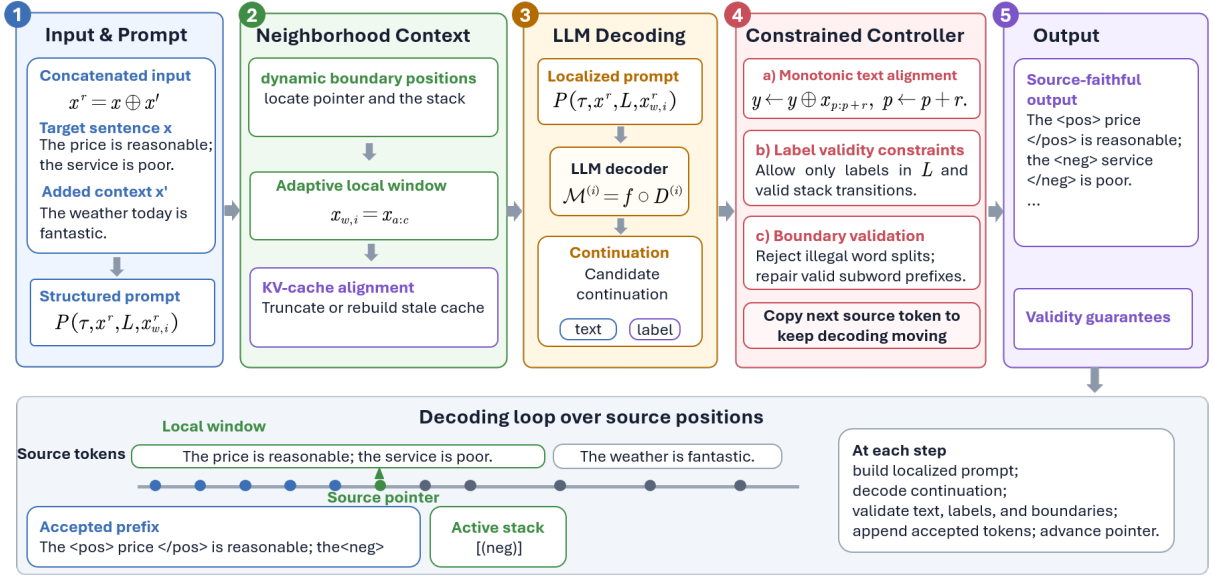


Figure 2: Overview of LoCoDec. Given a concatenated input $x^r = x \oplus x'$, LoCoDec decodes the output left-to-right. At each step, it selects a neighborhood context window, aligns the KV-cache, obtains text-and-label continuations from the LLM, and uses a deterministic controller to check text alignment, label validity, and boundary constraints. Accepted tokens are appended to the prefix until all source tokens are consumed, producing a source-faithful structured output y^r .

where $\ell^{(j)}$, $o^{(j)}$, and $s^{(j)}$ denote the start label, its output position, and its start position in the input sequence, respectively. We index the stack from the outermost to the innermost start label. Let $\alpha(s^{(1)})$ denote the starting position of the dynamic segment that contains $s^{(1)}$. If $\mathcal{S} \neq \emptyset$, we update

$$a \leftarrow \min(a, \alpha(s^{(1)})). \quad (4)$$

This adjustment prevents nested or long-span structures from being fragmented by superficial boundaries.

Output Neighborhood Window. The input neighborhood window restricts the visible prompt, but autoregressive decoding may still carry over cached attention states from previously processed tokens. For long input sequences, such stale states may still retain irrelevant evidence during local boundary determination, even if the input sequence is already restricted within the neighborhood. To ensure that the internal attention memory is consistent with the current local context, we align the KV cache before generation in each window.

Let K_l and V_l denote the cached key and value tensors at transformer layer l :

$$K_l, V_l \in \mathbb{R}^{B \times H \times T \times d}, \quad (5)$$

where B is the batch size, H is the number of attention heads, T is the cache length, and d is the head

dimension. Let κ_i denote the token length of the current localized prompt $P(\tau, x, L, x_{w,i})$ together with the accepted partial prefix $y_{w,i}$. If the existing cache corresponds to the same token prefix as the localized input, we truncate it as:

$$K'_l = K_l[:, :, : \kappa_i, :], \quad V'_l = V_l[:, :, : \kappa_i, :]. \quad (6)$$

The aligned cache is then:

$$C'_i = \{(K'_l, V'_l)\}_{l=1}^N, \quad (7)$$

where N is the number of transformer layers. When this prefix-alignment condition does not hold, we rebuild the cache from the localized prompt and the accepted partial prefix. This alignment removes stale attention states outside the current neighborhood context and makes subsequent decoding depend on the localized evidence and the accepted partial structure.

3.3 Constrained Token Paraphrasing

Monotonic Text Alignment. Under the neighborhood context strategy above, if the model output is not verified, the model may omit some input tokens, or generate tokens that do not exist in the input sequence x or the label set L . In addition, the start label and the end label may be generated in the wrong order.

Unlike methods that constrain the vocabulary step by step, we propose a segment-level verification strategy during decoding. Each time, the model generates a short segment, and LoCoDec decomposes this segment into a text segment and a candidate label segment.

To ensure output sequence fidelity, LoCoDec verifies the generated text segment. Specifically, this segment must align with the input sequence.

If a non-empty prefix can be aligned, LoCoDec copies the longest aligned text prefix into the output sequence and moves the pointer:

$$y \leftarrow y \oplus x_{p:p+r}, \quad p \leftarrow p + r. \quad (8)$$

For partial alignment, the unmatched suffix and subsequent candidate labels are discarded; candidate labels are validated only after full alignment.

If no non-empty prefix can be aligned, LoCoDec discards this segment and falls back to copying the next source token.

$$y \leftarrow y \oplus x_p, \quad p \leftarrow p + 1. \quad (9)$$

Label Validity Constraints. To ensure the validity of labels, the candidate label is accepted when it satisfies the following conditions: First, the label belongs to the predefined label set L ; second, if it is an end label, it must strictly match the current top label of the stack.

For a start label $\langle \ell \rangle$, LoCoDec first checks whether the label satisfies $\ell \in L$. If the check passes, LoCoDec appends the start label to the output sequence and pushes it, along with its positional metadata, onto the stack:

$$y \leftarrow y \oplus \langle \ell \rangle, \quad \mathcal{S} \leftarrow \text{push}(\mathcal{S}, (\ell, |y|, p)). \quad (10)$$

Here, $|y|$ denotes the current length of the output sequence, and p is the current source-text pointer. If $\ell \notin L$, the start label is strictly rejected.

For an end label $\langle / \ell \rangle$, LoCoDec checks whether the current stack is non-empty and determines whether the top label of the stack matches the end label. If the match succeeds, it appends the end label to the output sequence and pops the top label from the stack:

$$y \leftarrow y \oplus \langle / \ell \rangle, \quad \mathcal{S} \leftarrow \text{pop}(\mathcal{S}). \quad (11)$$

The end label is rejected if the stack is empty, the top label does not match ℓ , or $\ell \notin L$.

The stack-based constraint strategy maintains the validity of nested label sequences in a last-in, first-out manner and prevents the generation of invalid crossing structures.

Task	Dataset	Train	Valid	Test
NER	Weibo (Peng and Dredze, 2015)	1,350	270	270
	Resume (Zhang and Yang, 2018)	3,821	463	477
	W16 (Strauss et al., 2016)	2,394	1,000	3,850
	W17 (Derczynski et al., 2017)	3,394	1,008	1,287
NNER	GENIA (Wang et al., 2020)	15,023	1,669	1,854
	ACE2005 ¹ (Walker et al., 2006)	7,336	958	1,047
SF	ECDT ²	1,656	414	509
	CAIS (Liu et al., 2019)	7,995	994	1,024
ABSA	Laptop14 (Pontiki et al., 2014)	2,714	304	800
	Rest14 (Pontiki et al., 2014)	2,736	304	800
	Rest15 (Pontiki et al., 2015)	1,183	130	685
	Rest16 (Pontiki et al., 2016)	1,799	200	676
POS	CTB5 (Xue et al., 2005)	18,104	352	348
	CTB6 (Xue et al., 2005)	22,884	2,040	2,726
SRL	CN05 (Carreras and Màrquez, 2005)	90,750	3,248	5,267
	CPB1.0 (Xue, 2008)	17,839	1,115	2,018

Table 1: Statistics of the 16 datasets on the 6 evaluated structured prediction tasks. NER: Named Entity Recognition; NNER: Nested Named Entity Recognition; SF: Slot Filling; ABSA: Aspect-Based Sentiment Analysis; POS: Part-of-Speech; SRL: Semantic Role Labeling.

Boundary Checking and Lexical Repair.

LLMs may insert labels between two tokens that constitute a word. To avoid this problem, we adopt boundary checking and lexical repair methods to ensure that labels are not inserted inside a word. When the current token y_i generated by the LLM is a label, the previously generated token y_{i-1} is checked. If y_{i-1} is part of the word w at the corresponding position in the input sequence, then the currently generated label y_i is discarded, and the LLM is forced to output the subsequent tokens of w .

4 Experiment

4.1 Settings

Hyperparameters. The experiments are conducted on a Linux system running Ubuntu 16.04.7 LTS, utilizing NVIDIA A800 80GB PCIe GPUs for acceleration. All LLM-based methods are fine-tuned using LoRA (Hu et al., 2022) for parameter-efficient adaptation. During fine-tuning, LoRA rank is set to 8, batch size to 1, and the initial learning rate is set to 5×10^{-5} , with bf16 precision for training.

Dataset. We evaluate our method on sixteen benchmarks of six structured prediction tasks, as

¹<https://catalog.ldc.upenn.edu/LDC2006T06>

²<https://conference.cipsc.org.cn/smp2019/evaluation.html>

Method	S-F1	P-F1	R-F1	Tput.	S-F1	P-F1	R-F1	Tput.	S-F1	P-F1	R-F1	Tput.	S-F1	P-F1	R-F1	Tput.
<i>NER Tasks</i>	Weibo				Resume				W16				W17			
(Nie et al., 2020)	67.51	67.52	92.54	—	94.80	94.71	98.01	—	52.71	51.49	82.80	—	49.35	48.33	82.59	—
SSE	23.70	14.65	27.41	1.461	66.85	42.41	49.00	0.693	46.31	29.60	49.00	1.870	39.74	27.90	55.57	2.942
SCE	66.67	65.52	82.15	2.262	94.57	93.67	95.83	1.095	59.12	54.62	80.60	2.755	53.85	51.21	85.13	3.834
LoCoDec (Ours)	69.88	69.22	98.08	0.970	96.62	96.68	99.94	0.734	61.29	61.48	97.55	1.540	54.51	54.63	96.40	2.580
<i>ABSA Tasks</i>	Laptop14				Rest14				Rest15				Rest16			
(Chen et al., 2020)	65.87	63.39	89.99	—	76.64	75.64	90.95	—	67.11	66.24	89.87	—	73.61	74.21	92.39	—
SSE	61.29	40.88	50.83	2.315	66.13	42.19	50.52	1.429	51.85	30.95	39.89	2.324	63.93	37.67	47.66	2.284
SCE	71.49	63.75	82.63	3.166	76.89	72.01	83.40	2.043	72.64	70.67	80.27	3.253	80.34	76.01	84.76	3.097
LoCoDec (Ours)	72.59	73.36	97.34	2.027	82.13	82.11	99.19	1.452	76.40	77.23	98.40	2.263	81.71	81.46	97.96	2.286
<i>SF / SRL Tasks</i>	CAIS				ECDT				CPB1.0				CN05			
(Teng et al., 2021; Tian et al., 2022)	87.84	67.37	71.13	—	77.20	58.50	65.22	—	82.81	80.47	92.24	—	88.39	85.87	93.29	—
SSE	88.61	48.31	53.47	1.665	81.98	49.33	50.50	1.640	51.38	29.00	41.43	0.608	84.97	49.42	56.25	0.545
SCE	90.36	67.76	72.20	2.468	86.19	60.33	62.88	2.454	82.55	54.00	63.06	0.902	86.00	51.80	58.73	0.811
LoCoDec (Ours)	90.95	86.50	93.72	2.035	86.70	86.14	97.74	1.936	88.44	86.12	96.81	0.543	92.06	91.37	99.24	0.674
<i>POS / NNER Tasks</i>	CTB5				CTB6				GENIA				ACE2005			
(Tian et al., 2020; Yan et al., 2023)	96.34	96.42	98.88	—	94.82	94.97	98.53	—	80.27	80.33	96.52	—	84.05	84.12	96.25	—
SSE	68.42	46.91	50.30	0.129	64.64	43.28	47.74	0.116	71.43	47.89	56.81	0.733	72.54	55.84	62.14	0.581
SCE	92.34	90.68	92.77	0.243	90.09	86.61	88.93	0.213	77.46	74.97	87.33	1.082	82.83	80.36	86.93	0.849
LoCoDec (Ours)	95.77	95.74	99.75	0.226	93.81	93.35	99.62	0.164	79.61	79.78	98.73	0.846	87.96	87.95	97.66	0.668

Table 2: Main results across structured prediction benchmarks. LLM-based methods use the Llama3-8B backbone. We compare **LoCoDec** with two LLM-based output generation strategies, **SCE** and **SSE**, and with task-specific traditional baselines. **LoCoDec** inserts structural labels into paraphrased outputs, while SCE generates key-value pairs and SSE predicts span offsets. **Tput.** denotes inference throughput, measured as samples per second. Best results are highlighted in bold.

summarized in Table 1. Following Wang et al. (2021), we construct sentence-level robustness datasets by concatenating two adjacent sequences from the test set. The resulting double-sentence inputs preserve the semantics of the original sequences while introducing contextual distraction. To broaden the robustness analysis, we additionally consider five perturbation settings: topic drift, document-level noise, post-context noise, pre-context noise, and bidirectional context noise. Details of these settings and the complete results are provided in Appendix G.

Evaluation Metrics. We report three F1-based metrics for standard accuracy (S-F1), perturbed-input accuracy (P-F1), and prediction consistency (R-F1). Given $y = f(x)$, $y^r = f(x^r)$, and ground-truth labels y^t , the metrics are defined as

$$\begin{aligned}
\text{S-F1} &= \text{F1}(y, y^t), \\
\text{P-F1} &= \text{F1}(y^r, y^t), \\
\text{R-F1} &= \text{F1}(y^r, y).
\end{aligned} \tag{12}$$

Here, S-F1 evaluates the original input, P-F1 evaluates the perturbed input, and R-F1 measures prediction consistency. We retain P-F1 because R-F1 only measures agreement between the original and perturbed predictions and does not indicate whether

the perturbed prediction is correct. Hence, P-F1 complements R-F1 by measuring task accuracy after contextual expansion. Additional robustness analysis is provided in Appendix B.

Efficiency Metric. We report inference throughput (Tput.) as $\text{Tput.} = N_{\text{eval}}/T_{\text{inf}}$, where N_{eval} is the number of evaluated samples and T_{inf} is the total inference time. All LLM-based methods are evaluated on the same hardware and generation settings.

Backbone LLMs. We evaluate the proposed method on two representative LLMs, Llama3-8B³ (Dubey et al., 2024) and Qwen3-8B⁴ (Yang et al., 2025), to assess its effectiveness.

4.2 Main Results

Table 2 reports the main results with Llama3-8B. Additional Qwen3-8B results are provided in Appendix C.1, where the same overall trend is observed. We compare LoCoDec with two LLM-based generative baselines: Structured Content Extraction (SCE; generating structured key-value pairs) (Han et al., 2024) and Structured Span Extraction (SSE; predicting span offsets) (Ding et al.,

³<https://github.com/meta-llama/llama3>

⁴<https://github.com/QwenLM/Qwen3>

Method	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1
NER Tasks	Weibo			Resume			W16			W17		
- w/o all	68.08	68.07	90.86	96.62	95.67	97.41	60.80	59.86	90.04	51.53	51.89	86.71
- w/o nc	69.52	69.69	91.53	96.65	96.49	98.89	61.53	61.30	91.79	54.52	53.90	92.29
- w/o ctp	69.31	68.39	98.19	96.40	95.35	99.01	60.49	60.61	<u>97.09</u>	53.90	54.34	<u>95.94</u>
$w = k$	67.20	66.13	95.22	95.24	95.18	<u>99.38</u>	59.49	59.54	93.86	54.88	54.00	92.83
Ours	69.88	69.22	<u>98.08</u>	96.62	96.68	99.94	61.29	61.48	97.55	54.51	54.63	96.40
ABSA Tasks	Laptop14			Rest14			Rest15			Rest16		
- w/o all	72.43	70.41	90.42	78.63	78.91	90.72	73.66	70.70	84.42	79.80	78.42	87.37
- w/o nc	72.43	72.08	91.08	82.13	81.18	91.80	76.59	74.34	87.02	81.98	79.14	87.87
- w/o ctp	72.01	72.61	97.67	80.57	79.70	<u>99.15</u>	75.88	76.32	98.41	81.06	80.96	98.12
$w = k$	71.19	71.45	91.69	80.62	79.39	93.87	74.40	73.51	87.98	80.40	77.80	90.35
Ours	72.59	73.36	<u>97.34</u>	82.13	82.11	99.19	76.40	77.23	<u>98.40</u>	81.71	81.46	<u>97.96</u>
SF / SRL Tasks	CAIS			ECDT			CPB1.0			CN05		
- w/o all	90.14	79.13	83.23	85.67	74.45	81.37	87.21	74.49	79.91	90.45	79.40	83.88
- w/o nc	90.77	79.60	82.73	86.70	79.13	86.73	88.67	75.10	80.58	92.31	79.43	83.67
- w/o ctp	90.52	85.92	93.90	86.70	85.97	<u>97.59</u>	88.38	85.92	<u>96.71</u>	91.76	89.89	99.28
$w = k$	90.87	82.72	82.73	85.63	81.40	87.80	87.83	79.90	87.48	92.15	82.14	87.08
Ours	90.95	86.50	<u>93.72</u>	86.70	86.14	97.74	88.44	86.12	96.81	92.06	91.37	<u>99.24</u>
POS / NNER Tasks	CTB5			CTB6			GENIA			ACE2005		
- w/o all	94.59	94.13	98.56	92.35	90.37	96.76	79.13	79.40	92.66	86.81	86.46	92.20
- w/o nc	95.75	95.66	99.02	93.89	93.35	98.73	79.73	79.78	93.99	88.17	87.90	95.11
- w/o ctp	95.51	94.91	99.77	93.60	92.76	<u>99.57</u>	79.52	79.63	<u>98.46</u>	87.43	87.35	97.90
$w = k$	95.25	94.95	99.42	93.57	93.25	99.37	79.11	79.36	98.18	87.59	87.82	97.19
Ours	95.77	95.74	<u>99.75</u>	93.81	93.35	99.62	79.61	79.78	98.73	87.96	87.95	<u>97.66</u>

Table 3: Ablation results of LoCoDec using the Llama3-8B backbone. We compare the full method with four variants: (- w/o all), which removes both Neighborhood Context and Constrained Token Paraphrasing; (- w/o nc), which removes Neighborhood Context; (- w/o ctp), which removes Constrained Token Paraphrasing; and ($w = k$), which replaces the dynamic context with a fixed-radius window. Best results are highlighted in bold, and second-best R-F1 results are underlined.

2025). In addition, we provide supplementary controlled comparisons of LoCoDec with Liu et al. (2026), Ren and Teo (2025), and the Post-hoc baseline in Appendix E.

Comparison with Task-Specific Baselines. LoCoDec achieves competitive or better performance than task-specific models on most datasets. It shows clear advantages on ABSA and SRL, where aspect-level semantics and predicate-argument structures are important. It also performs robustly on social media datasets such as W16 and W17. However, LoCoDec still performs worse than specialized models on POS in S-F1 and P-F1. This is mainly because POS is a token-level task, and the labels are highly dense, while frequent insertion of labels makes the model more susceptible to the cumulative effect of autoregressive serialization errors. This indicates that LoCoDec provides a competitive unified framework with advantages on boundary-sensitive and semantically richer sequence-structured prediction tasks.

Accuracy and Robustness. LoCoDec achieves strong results with Llama3-8B on both standard accuracy and perturbation-based evaluation. Compared with SCE and SSE, LoCoDec generally improves perturbation robustness and prediction consistency while maintaining competitive S-F1. The gains are clear on representative datasets such as Weibo and Rest14, where LoCoDec preserves boundary recognition and prediction consistency under sentence-level perturbations. These results suggest that combining locality-aware context control with strict decoding constraints can reduce unstable span predictions and improve resistance to natural perturbations. The Qwen3-8B results in Appendix C.1 show a similar trend, further indicating that the effectiveness of LoCoDec is not limited to a single backbone.

Effect of Output Formulation. The weaker performance of SSE suggests that span-offset generation is a less stable output strategy for LLM-based sequence-structured prediction. Even small numerical deviations in predicted offsets can shift span boundaries and lead to boundary errors. While SCE

remains stronger than SSE on many datasets, it lacks direct surface-level alignment with the source text. In contrast, constrained token paraphrasing aligns naturally with the autoregressive generation process of LLMs. By sequentially reproducing text alongside labels, LoCoDec achieves faithful extraction without relying on absolute position regression. LoCoDec introduces additional constrained decoding, but its throughput remains close to or higher than SSE on several datasets. Further details on controller complexity and training cost are given in Appendix F.

4.3 Ablation Study

We conduct ablation studies to evaluate the main design choices of LoCoDec, as shown in Table 3. We compare the full framework with four variants: **- w/o all**, which removes both neighborhood context and constrained token paraphrasing; **- w/o nc**, which removes only neighborhood context; **- w/o ctp**, which removes only constrained token paraphrasing; and $w = k$, which uses a validation-selected fixed-radius context window instead of the adaptive sentence-level context window, with the selected values reported in Appendix B.3. Additional Qwen3-8B ablation results are provided in Appendix C.2, showing a similar overall trend.

Effect of Neighborhood Context. Neighborhood context is designed to suppress irrelevant long-range context caused by sentence-level perturbations. As shown by the comparison between the full model and **- w/o nc** in Table 3, removing this module consistently reduces R-F1 across all datasets, with the largest drop reaching 16.23 points. P-F1 also generally decreases or remains comparable, while the change in S-F1 is relatively small. These results indicate an asymmetric robustness-accuracy trade-off: local context isolation may discard some useful distant cues, but it yields substantially larger gains in perturbation stability. This phenomenon is related to the accuracy-robustness trade-off discussed by Tsipras et al. (2019). Fig. 3 and Appendix B provide further evidence that neighborhood context reduces sensitivity to distant contextual states.

Impact of Constrained Token Paraphrasing. The comparison between the full model and **- w/o ctp** directly isolates the effect of constrained token paraphrasing under the same neighborhood context setting. Removing this mechanism generally weakens S-F1 and P-F1, showing that un-

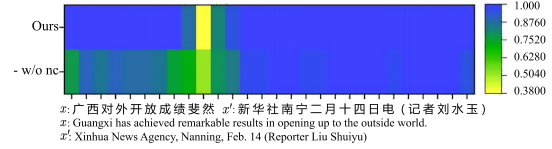


Figure 3: Cosine similarity of Qwen hidden representations between the original (x) and perturbed ($x' = x \oplus x'$) sequences. LoCoDec maintains highly consistent representations under contextual expansion, exhibiting superior stability.

constrained generation is more prone to boundary errors and surface-form mismatches. This effect is especially important for datasets where whitespace, punctuation, and multi-word expressions make surface alignment fragile. Although R-F1 sometimes remains comparable or even slightly higher after removing this module, R-F1 mainly measures prediction consistency rather than output correctness. Therefore, the main contribution of constrained token paraphrasing lies in improving source-text faithfulness, exact boundary matching, and perturbation accuracy. The comparison between **- w/o nc** and **- w/o all** further supports this observation under the setting without neighborhood context.

Benefits of Adaptive Context. We further compare the proposed adaptive context strategy with a fixed-radius window variant ($w = k$). The fixed-window variant often improves R-F1 over **- w/o nc**, confirming that restricting irrelevant context is useful. However, k is selected on the validation set, the fixed-window variant still underperforms the full LoCoDec framework on most robustness metrics. This suggests that the main limitation lies in the inflexibility of a fixed radius across heterogeneous tasks, rather than in the specific choice of k . LoCoDec instead uses an adaptive sentence-level window expanded according to the label stack ($\mathcal{S}$), preserving structural completeness while filtering irrelevant context. Different from the earlier span-step formulation, LoCoDec does not require a manually specified span step length during decoding. The decoding progress is determined by monotonic source-text alignment: LoCoDec moves by an accepted aligned text span when the alignment is reliable, and otherwise falls back to copying the next source token.

5 Conclusion

We propose LoCoDec, a locality-aware constrained decoding framework for LLM-based sequence-

structured prediction under long-context perturbations. LoCoDec formulates sequence-structured prediction tasks as constrained token paraphrasing and combines neighborhood context with strict decoding constraints. This design preserves the flexibility of autoregressive generation while improving source faithfulness and structural consistency. Experiments on sixteen benchmark datasets show that LoCoDec generally achieves stronger robustness performance than the two LLM-based output generation strategies, SCE and SSE, while remaining competitive with task-specific models on most datasets. The results suggest that locality-aware context control improves stability under sentence-level perturbations, while constrained token paraphrasing reduces boundary-sensitive generation errors. Overall, LoCoDec provides a competitive and robust framework for source-aligned sequence-structured prediction with LLMs, rather than a general solution for all forms of structured prediction.

Limitations

Although LoCoDec improves robustness in sequence-structured prediction, it has several limitations. First, its scope is restricted to source-aligned token, span, and nested-span outputs that can be serialized by inserting labels into the input sequence. It is not a general solution for arbitrary structured prediction, such as dependency parsing, AMR parsing, relation graphs, or document-level event graphs, unless these structures are first converted into a reliable sequence serialization. Second, its constrained decoding mechanism follows a greedy decision process. Without explicit lookahead, the decoder may accept spans that are locally plausible but globally inconsistent. Third, the design that limits access to long-distance text may reduce the accuracy of the model on complex tasks that require long-range dependency modeling or document-level reasoning. Fourth, our robustness evaluation focuses on sentence-level natural perturbations and does not fully cover adversarial perturbations, semantic-preserving rewrites, or document-level distribution shifts. Fifth, the dynamic window adjustment relies on deterministic heuristic rules rather than the model’s predictive uncertainty, which may reduce its adaptability in highly ambiguous contexts. In addition, string alignment and KV-cache manipulation require step-by-step interventions during decoding, increasing inference latency. Finally, as in many studies evalu-

ating pretrained LLMs, our results may be affected by data contamination. Potential exposure to benchmark datasets during large-scale pretraining could inflate performance and make comparisons with discriminative models trained from scratch less direct. From a broader-impact perspective, structural constraints cannot eliminate semantically incorrect predictions or bias propagation under context shifts. In sensitive domains such as those represented by GENIA and ACE2005, improved extraction robustness may also increase privacy and misuse risks when applied to private text streams. Future work will explore uncertainty-aware decoding and evaluate the framework on newly constructed, contamination-controlled benchmarks with more diverse robustness settings.

References

- Duarte Alves, Nuno Guerreiro, João Alves, José Pomal, Ricardo Rei, José de Souza, Pierre Colombo, and Andre Martins. 2023. [Steering large language models for machine translation with finetuning and in-context learning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11127–11148, Singapore. Association for Computational Linguistics.
- Xavier Carreras and Lluís Màrquez. 2005. [Introduction to the CoNLL-2005 shared task: Semantic role labeling](#). In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.
- Guimin Chen, Yuanhe Tian, and Yan Song. 2020. [Joint Aspect Extraction and Sentiment Analysis with Directional Graph Convolutional Networks](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 272–279, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Xiang Chen, Lei Li, Yuqi Zhu, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, Ningyu Zhang, and Huajun Chen. 2024a. [Sequence labeling as non-autoregressive dual-query set generation](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:1546–1558.
- Xuanting Chen, Junjie Ye, Can Zu, Nuo Xu, Tao Gui, and Qi Zhang. 2024b. [Robustness of GPT large language models on natural language processing tasks](#). *Journal of Computer Research and Development*, 61(5):1128–1142.
- Sudeshna Das and Jiaul Paik. 2022. [Resilience of named entity recognition models under adversarial attack](#). In *Proceedings of the First Workshop on Dynamic Adversarial Data Collection*, pages 1–6,

- Seattle, WA. Association for Computational Linguistics.
- Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. [Results of the WNUT2017 shared task on novel and emerging entity recognition](#). In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.
- Yifan Ding, Michael Yankoski, and Tim Weninger. 2025. [Span-oriented information extraction: A unified framework](#). *SIGKDD Explorations Newsletter*, 27(1):137–155.
- Ming Dong, Jinkui Zhang, Bolong Zheng, Xinhui Tu, Po Hu, and Tingting He. 2025. [DSCD: Large language model detoxification with self-constrained decoding](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3969–3984, Suzhou, China. Association for Computational Linguistics.
- Nathan Drenkow, Numair Sani, Ilya Shpitser, and Mathias Unberath. 2022. [A systematic review of robustness in Deep Learning for Computer Vision: Mind the gap?](#)
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. [LLaMA 3: The herd of models](#). *Preprint*, arXiv:2407.21783.
- Saibo Geng, Berkay Döner, Chris Wendler, Martin Josifoski, and Robert West. 2024. [Sketch-guided constrained decoding for boosting blackbox large language models without logit access](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 234–245, Bangkok, Thailand. Association for Computational Linguistics.
- Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, Jiafeng Guo, and Xueqi Cheng. 2024. [Retrieval-augmented code generation for universal information extraction](#). In *Natural Language Processing and Chinese Computing: Proceedings of NLPCC 2024, Part II*, pages 30–42, Hangzhou, China. Springer.
- Ridong Han, Chaohao Yang, Tao Peng, Prayag Tiwari, Xiang Wan, Lu Liu, and Benyou Wang. 2024. [An empirical study on information extraction using large language models](#). *Preprint*, arXiv:2305.14450.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *Proceedings of the International Conference on Learning Representations*.
- Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. [Evaluating open-domain question answering in the era of large language models](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada. Association for Computational Linguistics.
- Hongjin Kim, Jai-Eun Kim, and Harksoo Kim. 2024. [Exploring nested named entity recognition with large language models: Methods, challenges, and insights](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8653–8670, Miami, Florida, USA. Association for Computational Linguistics.
- Duong Minh Le, Yang Chen, Alan Ritter, and Wei Xu. 2024. [Constrained decoding for cross-lingual label projection](#). *Preprint*, arXiv:2402.03131.
- Guozheng Li, Wenjun Ke, Peng Wang, Zijie Xu, Ke Ji, Jiajun Liu, Ziyu Shang, and Qiqing Luo. 2024. [Unlocking instructive in-context learning with tabular prompting for relational triple extraction](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17131–17143, Torino, Italia. ELRA and ICCL.
- Bill Yuchen Lin, Wenyang Gao, Jun Yan, Ryan Moreno, and Xiang Ren. 2021. [RockNER: A simple method to create adversarial examples for evaluating the robustness of named entity recognition models](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3728–3737, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- C. Liu, L. Wang, P. Wang, and F. Jiang. 2026. [On sentence-level non-adversarial robustness of chinese named entity recognition with large language models](#). In *Natural Language Processing and Chinese Computing*, volume 16102 of *Lecture Notes in Computer Science*, pages 42–53, Singapore. Springer.
- Yijin Liu, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and Jinan Xu. 2019. [Cm-net: A novel collaborative memory network for spoken language understanding](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1051–1060, Hong Kong, China. Association for Computational Linguistics.
- Marcin Namysl, Sven Behnke, and Joachim Köhler. 2020. [NAT: Noise-aware training for robust neural sequence labeling](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1501–1517, Online. Association for Computational Linguistics.
- Yuyang Nie, Yuanhe Tian, Xiang Wan, Yan Song, and Bo Dai. 2020. [Named entity recognition for social media texts with semantic augmentation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1383–1391, Online. Association for Computational Linguistics.

- Nanyun Peng and Mark Dredze. 2015. [Named entity recognition for Chinese social media with jointly trained embeddings](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 548–554, Lisbon, Portugal. Association for Computational Linguistics.
- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. [Semeval-2016 task 5: Aspect based sentiment analysis](#). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.
- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. [Semeval-2015 task 12: Aspect based sentiment analysis](#). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado. Association for Computational Linguistics.
- Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Haris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. [Semeval-2014 task 4: Aspect based sentiment analysis](#). In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.
- Federico Raspanti, Tanir Ozcelebi, and Mike Holenderski. 2025. [Grammar-constrained decoding makes large language models better logical parsers](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 485–499, Vienna, Austria. Association for Computational Linguistics.
- Nicole Ren and James Teo. 2025. [SLENDER: Structured outputs for SLM-based NER in low-resource englishes](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 836–849, Vienna, Austria. Association for Computational Linguistics.
- Akshay Srinivasan and Sowmya Vajjala. 2023. [A multi-lingual evaluation of NER robustness to adversarial inputs](#). In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 40–53, Toronto, Canada. Association for Computational Linguistics.
- Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. 2016. [Results of the WNUT16 named entity recognition shared task](#). In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144, Osaka, Japan. The COLING 2016 Organizing Committee.
- Dechuan Teng, Libo Qin, Wanxiang Che, Sendong Zhao, and Ting Liu. 2021. [Injecting word information with multi-level word adapter for Chinese spoken language understanding](#). In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8188–8192. IEEE.
- Yuanhe Tian, Han Qin, Fei Xia, and Yan Song. 2022. [Syntax-driven approach for semantic role labeling](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7129–7139, Marseille, France. European Language Resources Association.
- Yuanhe Tian, Yan Song, and Fei Xia. 2020. [Joint Chinese word segmentation and part-of-speech tagging via multi-channel attention of character N-grams](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2073–2084, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Yuanhe Tian, Fei Xia, and Yan Song. 2024. [Large language models are no longer shallow parsers](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7131–7142, Bangkok, Thailand. Association for Computational Linguistics.
- Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. [Robustness may be at odds with accuracy](#). In *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*, New Orleans, LA, USA. OpenReview.net.
- Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. [ACE 2005 multilingual training corpus](#).
- Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. [Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 18647–18660.
- Jiayi Wang, Rongzhou Bao, Zhuosheng Zhang, and Hai Zhao. 2022. [Distinguishing non-natural from natural adversarial samples for more robust pre-trained language model](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 905–915, Dublin, Ireland. Association for Computational Linguistics.
- Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Haojun Huang, Wei Ye, Xiubo Geng, Binxin Jiao, Yue Zhang, and Xing Xie. 2023. [On the robustness of ChatGPT: An adversarial and out-of-distribution perspective](#).
- Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, Guoyin Wang, and Chen Guo. 2025. [GPT-NER: Named entity recognition via large language models](#). In *Findings of the*

Association for Computational Linguistics: NAACL 2025, pages 4257–4275, Albuquerque, New Mexico. Association for Computational Linguistics.

Yu Wang, Yun Li, Hanghang Tong, and Ziyi Zhu. 2020. [HIT: Nested named entity recognition via head-tail pair and token interaction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6027–6036, Online. Association for Computational Linguistics.

Nianwen Xue. 2008. [Labeling chinese predicates with semantic roles](#). *Computational Linguistics*, 34(2):225–255.

Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. [The penn chinese treebank: Phrase structure annotation of a large corpus](#). *Natural Language Engineering*, 11(2):207–238.

Hang Yan, Yu Sun, Xiaonan Li, and Xipeng Qiu. 2023. [An embarrassingly easy but strong baseline for nested named entity recognition](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1442–1452, Toronto, Canada. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. [Qwen3 technical report](#).

Sheng Yang, Min Li, and Luo Xu. 2024. [Named entity recognition based on large language model and instruction tuning](#). In *Proceedings of the 2024 5th International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*, pages 336–343.

Xuan Zhang, Navid Rajabi, Kevin Duh, and Philipp Koehn. 2023. [Machine translation with large language models: Prompting, few-shot learning, and fine-tuning with QLoRA](#). In *Proceedings of the Eighth Conference on Machine Translation*, pages 468–481, Singapore. Association for Computational Linguistics.

Yue Zhang and Jie Yang. 2018. [Chinese NER using lattice LSTM](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564, Melbourne, Australia. Association for Computational Linguistics.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Gong, and Xing Xie. 2024. [Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts](#). In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, pages 57–68, Salt Lake City, UT, USA. Association for Computing Machinery.

A Prompt Templates

In this section, we provide detailed prompt templates used in our experiments to support reproducibility. We present the actual instructions and formatting requirements for six representative sequence-structured prediction tasks: Slot Filling (SF), Named Entity Recognition (NER), Aspect-Based Sentiment Analysis (ABSA), Part-of-Speech (POS) tagging, Semantic Role Labeling (SRL), and Nested Named Entity Recognition (NNER).

A.1 Prompt Structure

As illustrated in Figure 1, LLM-based sequence-structured prediction in long-context scenarios often encounters structural hallucinations or boundary errors. To mitigate these issues, our templates are designed with the following core components:

- **Task Instruction:** Defines the specific extraction goal and introduces the target label set (e.g., *pos*, *neg*, *neu* for ABSA).
- **Context Integration:** Placeholders such as `{{Sequence to be annotated}}` are used to inject the input text and the dynamic local context window $x_{w,i}$ used by LoCoDec.
- **Few-shot Demonstration:** Provides concrete examples of the expected `<label>...</label>` format to guide the model’s generation logic.
- **Operational Requirements:** A set of explicit constraints (e.g., maintaining original semantics, following strict syntax) that complement the **LoCoDec** constrained decoding mechanism.

A.2 Task-Specific Templates

Figure 4 presents the task-specific prompt templates used in our experiments. The ABSA template illustrates how the model is guided to identify aspect terms and their corresponding sentiment polarities under a predefined structural schema. Similarly, the NER, POS, SF, and NNER templates adopt the same constrained generation format with task-specific label spaces, allowing us to maintain a unified prompting protocol across sequence-structured prediction benchmarks.

For SRL, we use a predicate-conditioned template because semantic role labeling depends on a given target predicate. Unlike the other tasks, each SRL input contains a pre-marked predicate in the form `$predicate$`. The model is then required to annotate only the semantic role arguments associated with this predicate, while keeping the predicate markers and the original text unchanged. Thus,

Aspect-Based Sentiment Analysis Task Prompt Template
This is an Aspect-Based Sentiment Analysis (ABSA) task. Please refer to the <example> and, based on the given <text> and <sentiment labels>, follow the <requirements> to extract aspect terms from the <text> and annotate their sentiment polarity. <Text> {Sequence to be annotated} <Sentiment Labels> {Sentiment labels set} <Example> Input: The price is reasonable although the service is poor . Output: The <pos> price <pos> is reasonable although the <neg> service <neg> is poor . <Requirements> 1. The provided <sentiment labels> are in the format <type>, representing the start and end labels for the corresponding sentiment polarity type; 2. The extraction process involves identifying aspect term expressions in the <text> that convey sentiment and marking them with the appropriate sentiment polarity labels; 3. Sentiment polarities include: pos (positive), neg (negative), neu (neutral); 4. The output format should follow the example output; 5. Ensure the original text's semantics and content remain unchanged, only adding sentiment polarity labels.

(a) ABSA

Named Entity Recognition Task Prompt Template
This is a named entity recognition task. Please refer to the <example> and extract entities from the given <text> based on the <entity labels> information, following the <requirements>. <Text> {Sequence to be annotated} <Entity Labels> {Entity label set} <Example> Input: November Rain Guns N ' Roses [URL] Output: <product> November Rain </product> <musicartist> Guns N ' Roses </musicartist> [URL] <Requirements> 1. The provided <entity labels> are in the form of <type>, representing the start and end labels for the entity type; 2. The extraction process involves determining whether a continuous fragment in the <text> belongs to a certain entity type, and if so, marking it with the corresponding labels; 3. The output format should refer to the output in the <example>; 4. Ensure that the semantics and content of the original text are not altered, only labels are added.

(b) NER

Part-of-Speech Tagging Task Prompt Template
这是一个词性标注任务，需要你参考<样例>，对给定的<文本>和<词性标签>信息，按<要求>，标注<文本>中每个词语的词性。 <文本> {待标记序列} <词性标签> {词性标签集合} <样例> 输入：残疾人体育事业也必将得到更广阔的发展空间。 输出：<nn> 残疾人 </nn><nn> 体育 </nn><nn> 事业 </nn><ad> 也 </ad><ad> 必 </ad><ad> 将 </ad><vv> 得到 </vv><ad> 更 </ad><va> 广阔 </va><dec> 的 </dec><nn> 发展 </nn><nn> 空间 </nn><pu>。 </pu> <要求> 1. 提供的<词性标签>为<type>，</type>形式，表示type词性对应的开始与结束标签； 2. 标注过程是识别<文本>中每个词语的词性，并用对应的词性标签进行标记； 3. 输出形式参考<样例>中的输出； 4. 确保原始文本的语义和内容不被改变，只添加词性标签。

(c) POS

Slot Filling Task Prompt Template
这是一个槽值填充任务，需要你参考<样例>，对给定的<文本>和<槽位标签>信息，按<要求>，提取<文本>中的槽位值。 <文本> {待标记序列} <槽位标签> {槽位标签集合} <样例> 输入：周杰伦的专辑《七里香》中的主打歌《我的地盘》是一首流行歌曲。 输出：<singer>周杰伦</singer>的专辑<album>《七里香》</album>中的主打歌<song>《我的地盘》</song>是一首<song_type>流行</song_type>歌曲。 <要求> 1. 提供的<槽位标签>为<type>，</type>形式，表示type槽位类型对应的开始与结束标签； 2. 提取过程是判断<文本>中某个连续的片段是否属于某个槽位类型，若是则用对应的标签进行标记； 3. 输出形式参考<样例>中的输出； 4. 确保原始文本的语义和内容不被改变，只添加标签。

(d) SF

Semantic Role Labeling Task Prompt Template
This is a semantic role labeling task. You need to refer to the <Example> and annotate the semantic roles in the given <text> based on the <Semantic Role Labels> information, following the <Requirements>. <Text> {Sequence to be annotated} <Semantic Role Labels> {Semantic role labels set} <Example> Input: Housing starts are Sexpected\$ to quicken a bit from August 's annual pace of 1,350,000 units . Output: <a1> Housing starts </a1> are Sexpected\$ </a1> to quicken a bit from August 's annual pace of 1,350,000 units </a1> . <Requirements> 1. The provided <Semantic Role Labels> are in the form of <role>, representing the start and end labels for the corresponding semantic role; 2. The annotation process is based on the predicate marked with \$ symbols in the text (already surrounded by \$), identifying the semantic role components (such as agent, patient, etc.) corresponding to this predicate, and marking them with the corresponding semantic role labels; 3. If the predicate has no corresponding semantic role components, no semantic role labels need to be added, just keep the original predicate markers; 4. The predicate is already marked with \$ symbols in the text (format: \$predicate\$), keep it as is in the output, no additional processing is needed; 5. Ensure that each semantic role component corresponding to the marked predicate is completely marked with the correct label, without omission, partial marking, or excessive marking; 6. Keep the original word order and content of the text unchanged, only add semantic role labels; 7. The output format strictly refers to the <Examples>, do not add extra spaces or line breaks between labels.

(e) SRL

Nested Named Entity Recognition Task Prompt Template
This is a Nested Named Entity Recognition (Nested NER) task. Please refer to the <example> and, based on the given <text> and <entity labels>, follow the <requirements> to extract named entities from the <text> and annotate them with their corresponding entity types. Note that entities can be nested within one another. <Text> {Sequence to be annotated} <Entity Labels> {Entity label set} <Example> Input: the police chief dine is in regular contact with the fbi . Output: <per> the <org> police </org> chief </per> <per> dine </per> is in regular contact with <org> the fbi </org> . <requirements> 1. The provided <entity labels> are in the format <type>, representing the start and end labels for the corresponding entity type; 2. The extraction process involves identifying named entities in the <text> and marking them with the appropriate entity type labels. Crucially, entities can be nested; 3. The specific entity types to extract will be provided in the <entity labels> list ; 4. The output format should strictly follow the example output, correctly nested labels in the proper order; 5. Ensure the original text's semantics, content, and punctuation remain unchanged, only adding the entity labels.

(f) NNER

Figure 4: Prompt templates for different sequence-structured prediction tasks in LLMs.

the SRL template remains consistent with constrained token paraphrasing, but introduces predicate marking to specify the target predicate before decoding.

B Robustness Analysis

B.1 LLM Decoding

Following the task formulation in Section 3.1, LoCoDec performs autoregressive decoding with the dynamic prompt $P(\tau, x, L, x_{w,i})$ and the output neighborhood window $y_{w,i}$.

To model this stepwise generation process explicitly, we decompose the overall model operation at step i , denoted as $\mathcal{M}^{(i)}$, into two stages: hidden state representation and token projection.

First, the decoder backbone, denoted as $D^{(i)}$, maps the current context, which comprises $P(\tau, x, L, x_{w,i})$ and the output neighborhood window $y_{w,i}$, to a dense hidden representation $\mathbf{v}_i$:

$$\mathbf{v}_i = D^{(i)}(P(\tau, x, L, x_{w,i}) \oplus y_{w,i}). \quad (13)$$

A decoding head $f(\cdot)$ (e.g., softmax) then generates the output token: $y_i \sim f(\mathbf{v}_i)$.

Thus, the complete model operation corresponds to the composition of these components: $\mathcal{M}^{(i)} = f \circ D^{(i)}$. Accordingly,

$$y_i = \mathcal{M}^{(i)}(P(\tau, x, L, x_{w,i}) \oplus y_{w,i}). \quad (14)$$

We define the output sequence as $y = \bigoplus_{i=1}^m y_i$, where each token $y_i \in \{x_1, \dots, x_n\} \cup \mathcal{T}(L)$. Here, $\mathcal{T}(L) = \{\langle \ell \rangle, \langle / \ell \rangle \mid \ell \in L\}$ represents the structural label set derived from L , and m is the total number of decoded tokens.

B.2 Robustness Metric

We define the Prediction Inconsistency Rate (PIR) to measure the robustness of sequence-structured prediction models under sentence-level natural perturbations in long-context settings. PIR quantifies the proportion of token-level predictions that change after perturbation.

Formally, given an original input sequence x and

its perturbed counterpart x^r , PIR is defined as

$$\text{PIR} = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(f(\mathbf{v}_i) \neq f(\mathbf{v}_i^r)), \quad (15)$$

where $\mathbf{v}_i$ and $\mathbf{v}_i^r$ denote the hidden representations at position i when decoding x and x^r , respectively.

Sentence-level perturbations introduce additional contextual information during context processing, which may shift hidden representations and alter predictions. In particular, prediction changes are likely when the representation similarity drops below a threshold, which can be characterized by cosine similarity and Euclidean distance:

$$\cos(\mathbf{v}_i, \mathbf{v}_i^r) < 1 - \varepsilon, \quad \|\mathbf{v}_i - \mathbf{v}_i^r\| > \delta, \quad (16)$$

where ε and δ control the perturbation strength.

The hidden representations are produced by the decoder conditioned on a structured prompt and the previously generated prefix:

$$\begin{aligned} \mathbf{v}_i &= \mathcal{D}^{(i)}(P(\tau, x, L, x_{w,i}) \oplus y_{w,i}), \\ \mathbf{v}_i^r &= \mathcal{D}^{(i)}(P(\tau, x^r, L, x_{w,i}^r) \oplus y_{w,i}^r). \end{aligned} \quad (17)$$

When a perturbation is located outside the active local context window, the localized context remains identical ($x_{w,i} = x_{w,i}^r$ and $y_{w,i} = y_{w,i}^r$). Crucially, reinforced by our KV-cache truncation mechanism that masks out distant stale attention states, the representations become less sensitive to global prompt variations, yielding:

$$\mathbf{v}_i \approx \mathbf{v}_i^r \Rightarrow f(\mathbf{v}_i) \approx f(\mathbf{v}_i^r) \quad (18)$$

This observation suggests that local structural dependencies are more resilient to distant context noise than global semantic representations.

B.3 Fixed-Window Radius

For the fixed-window ablation variant, we select the window radius k on the validation set for each dataset. The selected value is fixed during test evaluation. Table 4 reports the selected values.

C Additional Results with Qwen3-8B

To further examine whether the effectiveness of LoCoDec depends on a specific backbone model, we report additional results using Qwen3-8B. The main text presents Llama3-8B as the primary backbone for readability, while the complete Qwen3-8B results are provided in this appendix.

Task	Dataset	Llama3-8B	Qwen3-8B
NER	Weibo	12	10
	Resume	12	10
	W16	10	10
	W17	14	10
NNER	GENIA	14	14
	ACE2005	14	14
SF	ECDT	8	8
	CAIS	8	8
ABSA	Laptop14	10	12
	Rest14	14	12
	Rest15	14	14
	Rest16	15	14
POS	CTB5	8	8
	CTB6	10	10
SRL	CN05	45	45
	CPB1.0	45	40

Table 4: Validation-selected fixed-window radius k for the fixed-window ablation variant. For each dataset and backbone, k is selected on the validation set and then fixed during test evaluation.

C.1 Main Results with Qwen3-8B

As shown in Table 5, LoCoDec shows a similar trend to the Llama3-8B results reported in the main text. Compared with the generative baselines SCE and SSE, LoCoDec generally achieves stronger perturbation robustness and higher prediction consistency across different sequence-structured prediction tasks. These results suggest that the improvements of LoCoDec are not limited to a single LLM backbone.

C.2 Ablation Study with Qwen3-8B

Table 6 reports the ablation results with Qwen3-8B. Removing all proposed components leads to clear performance degradation, while removing neighborhood context mainly affects perturbation robustness. The fixed-window variant improves over the model without neighborhood context, but it still underperforms the full LoCoDec framework on robustness-oriented metrics. This further supports the effectiveness of dynamic locality control and constrained token paraphrasing.

D Segment-Level Validation Controller

Algorithm 1 details the segment-level validation controller that enforces source alignment, label validity, and structural consistency in LoCoDec.

Method	S-F1	P-F1	R-F1	Tput.	S-F1	P-F1	R-F1	Tput.	S-F1	P-F1	R-F1	Tput.	S-F1	P-F1	R-F1	Tput.
<i>NER Tasks</i>	Weibo				Resume				W16				W17			
(Nie et al., 2020)	67.51	67.52	92.54	–	94.80	94.71	98.01	–	52.71	51.49	82.80	–	49.35	48.33	82.59	–
SSE	24.54	12.42	26.02	0.950	66.01	43.42	48.30	0.460	39.93	25.03	41.32	1.611	33.74	23.17	49.03	2.405
SCE	65.32	64.44	81.52	1.852	96.17	93.00	95.38	0.912	56.52	53.22	80.17	2.413	52.35	51.37	84.86	3.197
LoCoDec (Ours)	68.82	68.60	98.20	0.817	96.72	96.69	99.97	0.519	59.93	59.79	96.42	1.346	53.69	54.02	96.50	2.125
<i>ABSA Tasks</i>	Laptop14				Rest14				Rest15				Rest16			
(Chen et al., 2020)	65.87	63.39	89.99	–	76.64	75.64	90.95	–	67.11	66.24	89.87	–	73.61	74.21	92.39	–
SSE	59.27	39.71	52.39	1.795	63.77	40.78	49.82	1.178	49.47	28.11	35.66	1.990	58.01	37.28	42.31	2.030
SCE	72.80	67.88	81.26	2.555	75.71	71.84	83.47	1.658	73.66	69.25	79.56	2.502	77.64	75.99	83.18	2.723
LoCoDec (Ours)	72.01	72.61	97.67	1.801	80.61	80.43	98.73	1.158	73.70	72.38	97.50	1.859	80.16	79.38	98.09	1.817
<i>SF / SRL Tasks</i>	CAIS				ECDT				CPB1.0				CN05			
(Teng et al., 2021; Tian et al., 2022)	87.84	67.37	71.13	–	77.20	58.50	65.22	–	82.81	80.47	92.24	–	88.39	85.87	93.29	–
SSE	91.55	52.24	51.07	1.280	82.34	49.40	52.49	1.349	52.36	30.08	43.55	0.488	83.19	49.27	56.74	0.405
SCE	92.43	70.59	70.11	2.278	87.72	66.21	69.07	2.105	83.05	54.22	62.61	0.716	85.44	51.32	58.78	0.657
LoCoDec (Ours)	92.87	92.63	98.99	1.387	87.69	86.53	98.50	1.360	88.38	88.27	99.94	0.479	92.05	90.70	99.10	0.480
<i>POS / NNER Tasks</i>	CTB5				CTB6				GENIA				ACE2005			
(Tian et al., 2020; Yan et al., 2023)	96.34	96.42	98.88	–	94.82	94.97	98.53	–	80.27	80.33	96.52	–	84.05	84.12	96.25	–
SSE	63.02	43.49	45.23	0.096	63.74	44.37	44.77	0.087	67.59	42.83	51.19	0.548	68.17	50.34	56.97	0.457
SCE	92.43	89.42	87.65	0.231	89.95	87.12	81.97	0.217	76.27	73.70	86.47	0.914	81.06	67.42	84.24	0.760
LoCoDec (Ours)	95.94	95.80	99.77	0.150	94.23	94.23	99.77	0.113	79.75	80.15	98.55	0.631	86.23	86.54	96.61	0.524

Table 5: Main results across structured prediction benchmarks. LLM-based methods use the Qwen3-8B backbone. We compare **LoCoDec** with two LLM-based output generation strategies, **SCE** and **SSE**, and with task-specific traditional baselines. **LoCoDec** inserts structural labels into paraphrased outputs, while SCE generates key-value pairs and SSE predicts span offsets. **Tput.** denotes inference throughput, measured as samples per second. Best results are highlighted in bold.

Method	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1
NER Tasks	Weibo			Resume			W16			W17		
- w/o all	67.98	67.82	87.32	96.51	93.93	97.91	59.71	58.53	87.12	53.12	50.82	86.64
- w/o nc	69.06	68.61	87.48	96.72	95.59	<u>98.70</u>	60.24	59.31	87.76	53.77	51.80	86.66
- w/o ctp	68.40	68.15	98.35	96.00	95.50	98.30	58.71	58.64	<u>96.18</u>	53.02	52.73	96.79
$w = k$	66.49	67.20	94.78	95.71	94.66	98.37	59.47	59.57	94.26	52.73	51.79	93.78
Ours	68.82	68.60	<u>98.20</u>	96.72	96.69	99.97	59.93	59.79	96.42	53.69	54.02	<u>96.50</u>
ABSA Tasks	Laptop14			Rest14			Rest15			Rest16		
- w/o all	70.05	70.10	86.03	79.98	78.12	89.70	72.33	68.99	85.54	78.49	77.77	89.47
- w/o nc	70.16	70.12	87.01	80.61	79.22	91.03	73.49	70.77	86.58	80.17	79.35	88.98
- w/o ctp	70.10	71.05	98.47	80.50	80.07	98.78	72.43	71.42	97.53	80.03	78.88	<u>97.90</u>
$w = k$	70.39	71.11	92.80	79.78	78.58	93.84	72.41	69.22	90.93	78.80	78.23	91.82
Ours	72.01	72.61	<u>97.67</u>	80.61	80.43	<u>98.73</u>	73.70	72.38	<u>97.50</u>	80.16	79.38	98.09
SF / SRL Tasks	CAIS			ECDT			CPB1.0			CN05		
- w/o all	92.79	79.75	82.49	87.56	71.34	75.82	88.34	75.40	80.50	89.36	80.42	85.09
- w/o nc	92.79	80.51	82.88	87.69	72.96	78.21	88.41	75.73	81.40	92.09	80.66	85.31
- w/o ctp	92.70	92.42	<u>98.78</u>	87.59	86.40	<u>97.61</u>	88.32	88.01	<u>99.78</u>	91.16	88.75	<u>99.07</u>
$w = k$	92.51	82.72	85.21	86.06	70.30	80.13	88.05	79.88	87.19	91.99	82.15	87.10
Ours	92.87	92.63	98.99	87.69	86.53	98.50	88.38	88.27	99.94	92.05	90.70	99.10
POS / NNER Tasks	CTB5			CTB6			GENIA			ACE2005		
- w/o all	95.35	95.10	99.07	93.56	92.69	97.91	79.25	78.87	92.81	85.81	85.40	92.48
- w/o nc	95.92	95.73	99.18	94.33	94.20	98.88	79.66	79.93	93.25	86.39	85.59	93.79
- w/o ctp	95.41	95.24	99.92	93.41	93.00	99.81	79.42	79.81	97.82	86.06	85.79	97.46
$w = k$	95.26	94.93	99.43	93.63	93.43	99.55	79.13	79.17	<u>98.22</u>	86.38	86.42	<u>96.67</u>
Ours	95.94	95.80	<u>99.77</u>	94.23	94.23	<u>99.77</u>	79.75	80.15	98.55	86.23	86.54	96.61

Table 6: Ablation results of the LoCoDec framework using the Qwen3-8B backbone. We compare the full method with four variants: (- w/o all), which removes both Neighborhood Context and Constrained Token Paraphrasing; (- w/o nc), which removes Neighborhood Context; (- w/o ctp), which removes Constrained Token Paraphrasing; and ($w = k$), which replaces the dynamic sentence-level context with a validation-selected fixed-radius window. Best results are highlighted in bold, and second-best R-F1 scores are underlined.

Algorithm 1 Segment-level Validation Controller

Input: source tokens X , model M , legal labels L **Output:** validated structured output Y

```
1:  $Y \leftarrow \text{empty}$ ;  $i \leftarrow 0$ ;  $S \leftarrow \text{empty stack}$ ;  $KV \leftarrow \text{PREFIX}(M)$ 
2: while  $i$  is not at the end of  $X$  do
3:    $W \leftarrow \text{SENTENCE\_WINDOW}(X, i, S)$   $\triangleright$  CORE: local sentence window
4:    $n \leftarrow \max(25, \min(W.\text{right} - i + 25, 60))$   $\triangleright$  Bound each decoding call
5:    $C \leftarrow \text{GREEDY\_DECODE}(M, Y, KV, W, n)$   $\triangleright$  Generate one validation unit
6:    $(T, G) \leftarrow \text{SPLIT\_TEXT\_AND\_LABELS}(C)$   $\triangleright$  Text prefix  $T$  and candidate labels  $G$ 
7:    $(P, \text{state}) \leftarrow \text{ALIGN\_LONGEST\_PREFIX}(T, X, i)$   $\triangleright$  CORE: NONE, PARTIAL, or FULL
8:   if  $\text{state} = \text{NONE}$  then
9:      $Y \leftarrow Y + X[i]$ ;  $i \leftarrow i + 1$   $\triangleright$  FALLBACK: force exact source token
10:     $KV \leftarrow \text{RESTORE\_TO\_VERIFIED\_PREFIX}(M, KV, Y)$   $\triangleright$  ROLLBACK: remove unverified states
11:    continue
12:   end if
13:    $Y \leftarrow Y + P$ ;  $i \leftarrow i + |P|$   $\triangleright$  CORE: accept aligned prefix
14:    $\text{progressed} \leftarrow (|P| > 0)$ 
15:   if  $\text{state} = \text{FULL}$  then  $\triangleright$  Validate labels only after full alignment
16:     for all  $g \in G$  do
17:       if  $\text{VALID\_TYPE}(g, L)$  and  $\text{VALID\_BOUNDARY}(g, X, i)$  and  $\text{VALID\_STACK}(g, S)$  then  $\triangleright$  CORE: label
18:         type, boundary, and nesting
19:          $Y \leftarrow Y + g$ ;  $\text{UPDATE\_STACK}(S, g)$ 
20:          $\text{progressed} \leftarrow \text{true}$ 
21:       end if
22:     end for
23:   end if  $\triangleright$  PARTIAL: discard unmatched suffix and labels
24:   if not  $\text{progressed}$  then
25:      $Y \leftarrow Y + X[i]$ ;  $i \leftarrow i + 1$   $\triangleright$  FALLBACK: guarantee monotonic progress
26:   end if
27:    $KV \leftarrow \text{RESTORE\_TO\_VERIFIED\_PREFIX}(M, KV, Y)$   $\triangleright$  ROLLBACK: synchronize verified cache
28: end while
29:  $Y \leftarrow \text{CLOSE\_OPEN\_LABELS}(Y, S)$   $\triangleright$  Ensure a well-formed label structure
30: return  $Y$ 
```

D.1 Candidate Decomposition and Alignment

A candidate segment is decomposed into a text part followed by contiguous candidate labels. Starting from the current text-alignment pointer, the controller searches for the longest aligned prefix between the generated text and the source sequence via `ALIGN_LONGEST_PREFIX`. If a non-empty aligned prefix exists, the controller accepts the aligned part of the generated text and advances the pointer accordingly; if the generated text is only partially aligned, the unaligned part and all subsequent candidate labels are discarded. Candidate labels are validated against label type, language boundary, and nesting-stack constraints only when the generated text is fully aligned. If no non-empty aligned prefix exists, the next exact source token is forced as output.

D.2 Repair and KV-Cache Synchronization

For unalignable or invalid candidates, the controller discards the erroneous generation and truncates or rebuilds the KV cache to the verified prefix. This mechanism is implemented by re-running greedy decoding after prefix repair, and it uses no vocabulary masking and does not directly modify the

next-token logits.

D.3 Boundary Validation and Subword Repair

English word boundaries are detected by whitespace and punctuation and labels must not split contiguous letters or digits, while Chinese uses character boundaries via exact alignment with the source text. For subword tokenizers, if a label boundary falls inside a source token, the controller splits the token and re-encodes the two parts so the label is validated at the correct character boundary. This relies on no hard-coded token IDs and adapts to different subword tokenizers such as BPE and SentencePiece.

E Extended Comparative Evaluation

To further assess the effectiveness of LoCoDec, we provide an extended comparison with [Liu et al. \(2026\)](#), a fixed-window LoCoDec variant, a post-hoc constraint baseline, and [Ren and Teo \(2025\)](#) on all 16 structured prediction datasets. All methods use the same Llama3-8B backbone, data splits, and evaluation protocol.

The method of [Liu et al. \(2026\)](#) uses the same

Method	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1
NER Tasks												
	Weibo			Resume			W16			W17		
(Liu et al., 2026)	66.49	66.75	95.54	95.85	96.01	98.19	59.28	59.50	93.87	53.35	52.71	93.71
LoCoDec ($w = k$)	67.20	66.13	95.22	95.24	95.18	99.38	59.49	59.54	93.86	54.88	54.00	92.83
Post-hoc	69.77	68.65	91.41	96.43	96.55	99.88	60.81	61.12	97.25	53.90	54.49	95.94
(Ren and Teo, 2025)	67.56	66.17	85.79	92.80	91.42	98.32	61.22	61.16	88.33	54.25	52.34	88.34
LoCoDec (Ours)	69.88	69.22	98.08	96.62	96.68	99.94	61.29	61.48	97.55	54.51	54.63	96.40
ABSA Tasks												
	Laptop14			Rest14			Rest15			Rest16		
(Liu et al., 2026)	70.06	69.19	92.12	80.82	79.61	93.89	73.85	71.78	87.72	79.44	77.38	91.08
LoCoDec ($w = k$)	71.19	71.45	91.69	80.62	79.39	93.87	74.40	73.51	87.98	80.40	77.80	90.35
Post-hoc	72.76	70.73	91.58	81.64	81.57	99.15	73.75	72.71	86.34	80.62	79.41	87.37
(Ren and Teo, 2025)	72.83	71.77	88.13	80.46	79.07	89.67	74.53	70.59	85.34	79.01	75.93	86.80
LoCoDec (Ours)	72.59	73.36	97.34	82.13	82.11	99.19	76.40	77.23	98.40	81.71	81.46	97.96
SF / SRL Tasks												
	CAIS			ECDT			CPB1.0			CN05		
(Liu et al., 2026)	89.56	83.47	83.64	86.36	80.00	84.40	86.34	80.06	87.81	92.20	82.05	86.80
LoCoDec ($w = k$)	90.87	82.72	82.73	85.63	81.40	87.80	87.83	79.90	87.48	92.15	82.14	87.08
Post-hoc	90.14	80.08	83.70	86.70	86.13	97.58	88.38	86.10	96.73	82.41	71.85	83.80
(Ren and Teo, 2025)	90.02	73.42	75.28	87.02	76.04	78.16	88.38	69.43	73.78	92.04	71.99	79.83
LoCoDec (Ours)	90.95	86.50	93.72	86.70	86.14	97.74	88.44	86.12	96.81	92.06	91.37	99.24
POS / NNER Tasks												
	CTB5			CTB6			GENIA			ACE2005		
(Liu et al., 2026)	95.21	94.91	99.45	93.61	93.28	99.31	–	–	–	–	–	–
LoCoDec ($w = k$)	95.25	94.95	99.42	93.57	93.25	99.37	79.11	79.36	98.18	87.59	87.82	97.19
Post-hoc	95.71	95.72	99.67	93.76	93.24	98.44	79.32	80.04	93.32	87.69	87.43	97.59
(Ren and Teo, 2025)	94.67	93.61	98.86	93.34	93.28	99.58	78.84	77.34	91.37	86.43	83.59	93.58
LoCoDec (Ours)	95.77	95.74	99.75	93.81	93.35	99.62	79.61	79.78	98.73	87.96	87.95	97.66

Table 7: Comparison of Liu et al. (2026), LoCoDec with a fixed window ($w = k$), Ren and Teo (2025), Post-hoc, and LoCoDec (Ours) on the 16 datasets. All LLM-based methods use the Llama3-8B backbone. “–” denotes an unavailable result, and best results are highlighted in bold.

fixed-window strategy as LoCoDec ($w = k$) variant, thereby isolating the contribution of adaptive context construction. The Post-hoc baseline first generates unconstrained outputs and subsequently performs structural repair, including illegal-label filtering, stack-based label matching, removal of isolated labels, and removal of empty labels. The method of Ren and Teo (2025) is reproduced using its structured pair-list output formulation because its official implementation is unavailable.

Liu et al. (2026) was originally designed for flat Chinese NER. The English and non-NER results come from our cross-lingual and cross-task transfer of the method of Liu et al. (2026). A result marked by “–” indicates that the method cannot be directly applied under the corresponding task structure.

As shown in Table 7, LoCoDec ($w = k$), which adopts the same fixed window as Liu et al. (2026), achieves higher scores than Liu et al. (2026) on several datasets, suggesting the benefit of segment-level validation over per-token filtering. The full LoCoDec further improves upon this variant, supporting the contribution of adaptive context construction. Overall, LoCoDec generally achieves higher P-F1 and R-F1 while maintaining competi-

tive S-F1 against the other baselines.

F Complexity and Training Cost

Controller Complexity. We analyze the computational cost of LoCoDec separately from autoregressive language-model decoding. Excluding language-model decoding, the conservative time complexity of the controller is

$$O(N_x + K_{\text{val}}B_{\text{max}}^2 + Z_{\text{lab}}), \quad (19)$$

where N_x , K_{val} , B_{max} , and Z_{lab} denote the source length, the number of validation calls, the candidate-length cap, and the total number of candidate labels, respectively. Since $B_{\text{max}} \leq 60$ and no beam search or vocabulary-level logit masking is used, the additional computation grows approximately linearly with N_x and K_{val} .

Training Cost. LoCoDec adds no training objectives or trainable modules, so it incurs no extra training cost. Table 8 reports the training epochs and measured training time for Llama3-8B and Qwen3-8B on the 16 datasets.

Task	Dataset	Epochs (L/Q)	Llama3-8B (h)	Qwen3-8B (h)
NER	Weibo	4 / 4	1.00	1.10
	Resume	2 / 2	1.42	1.55
	W16	4 / 4	2.15	3.90
	W17	4 / 4	5.62	5.61
NNER	GENIA	4 / 4	16.39	18.20
	ACE2005	4 / 4	5.21	7.88
SF	ECDT	6 / 4	2.79	1.67
	CAIS	4 / 4	9.14	6.51
ABSA	Laptop14	4 / 4	4.26	2.20
	Rest14	4 / 4	2.02	2.22
	Rest15	4 / 4	0.88	0.97
	Rest16	4 / 4	1.33	1.47
POS	CTB5	2 / 2	7.86	8.70
	CTB6	2 / 2	7.71	15.09
SRL	CN05	2 / 2	39.04	39.26
	CPB1.0	2 / 2	5.31	5.84

Table 8: Training epochs and training time for Llama3-8B and Qwen3-8B on 16 datasets. Training time is reported in hours, and “L/Q” denotes Llama3-8B/Qwen3-8B.

G Extended Robustness Evaluation

We consider five perturbation settings. *Topic drift* concatenates each target instance with a randomly sampled distractor sequence from a different dataset or domain, thereby introducing a topical or domain shift. *Document-level noise* augments the target with irrelevant document-level elements, such as titles, dates, section markers, page numbers, and text resembling headers or footers. To isolate the effect of context position, we further construct *post-context noise*, *pre-context noise*, and *bidirectional context noise* by placing distractor sentences after the target, before the target, or on both sides of the target, respectively.

Table 9 presents the complete results. Overall, LoCoDec generally achieves higher P-F1 and R-F1 than the - **w/o all** ablation under most perturbation settings, although the magnitude of the improvement varies with different datasets and task types. The position-controlled experiments suggest that preceding and bidirectional distractors are more challenging for the ablation on several datasets, while LoCoDec generally maintains higher perturbed-input accuracy and prediction consistency. These results suggest that LoCoDec is less sensitive to changes in contextual content and position.

<i>NER Tasks</i>	Method	Weibo			Resume			W16			W17		
		S-F1	P-F1	R-F1	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1
Topic drift	- w/o all	68.08	67.92	92.96	96.62	96.40	99.44	60.80	60.96	95.99	51.53	51.97	90.58
	LoCoDec (Ours)	69.88	68.55	94.42	96.62	96.55	99.94	61.29	61.37	98.16	54.51	54.23	98.21
Document-level noise	- w/o all	68.08	68.12	89.13	96.62	95.93	98.42	60.80	59.65	90.76	51.53	51.55	90.83
	LoCoDec (Ours)	69.88	69.83	95.62	96.62	96.49	99.81	61.29	61.36	98.75	54.51	53.86	98.45
Post-context noise	- w/o all	68.08	73.95	95.00	96.62	96.58	98.91	60.80	61.37	94.91	51.53	55.04	92.80
	LoCoDec (Ours)	69.88	68.25	98.35	96.62	97.20	99.86	61.29	62.33	98.82	54.51	57.40	98.16
Pre-context noise	- w/o all	68.08	62.37	87.09	96.62	94.88	96.33	60.80	59.59	85.60	51.53	49.80	82.93
	LoCoDec (Ours)	69.88	68.53	99.73	96.62	96.06	99.87	61.29	60.64	96.34	54.51	51.48	94.16
Bidirectional context noise	- w/o all	68.08	66.97	83.87	96.62	97.28	97.64	60.80	60.16	85.85	51.53	50.46	84.55
	LoCoDec (Ours)	69.88	69.49	90.76	96.62	97.92	99.78	61.29	61.78	96.31	54.51	54.86	94.69

<i>ABSA Tasks</i>	Method	Laptop14			Rest14			Rest15			Rest16		
		S-F1	P-F1	R-F1	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1
Topic drift	- w/o all	72.43	69.57	93.27	78.63	77.19	94.93	73.66	69.07	86.37	79.80	76.13	90.72
	LoCoDec (Ours)	72.59	72.43	99.82	82.13	81.95	99.73	76.40	76.05	98.52	81.71	81.48	99.41
Document-level noise	- w/o all	72.43	69.95	91.15	78.63	79.62	93.61	73.66	72.76	93.26	79.80	76.83	90.27
	LoCoDec (Ours)	72.59	73.17	97.69	82.13	81.86	98.48	76.40	75.21	97.44	81.71	81.92	98.39
Post-context noise	- w/o all	72.43	71.29	93.62	78.63	79.85	94.68	73.66	71.95	90.16	79.80	79.21	91.54
	LoCoDec (Ours)	72.59	73.99	99.86	82.13	83.73	99.82	76.40	76.12	99.80	81.71	82.03	99.52
Pre-context noise	- w/o all	72.43	70.19	89.22	78.63	79.61	87.26	73.66	72.14	80.63	79.80	78.61	82.39
	LoCoDec (Ours)	72.59	72.73	94.58	82.13	80.53	98.58	76.40	78.33	97.04	81.71	80.83	96.23
Bidirectional context noise	- w/o all	72.43	69.83	85.99	78.63	83.08	85.23	73.66	69.03	76.60	79.80	76.30	80.10
	LoCoDec (Ours)	72.59	72.38	94.38	82.13	86.15	97.89	76.40	75.00	93.46	81.71	85.31	97.17

<i>SF / SRL Tasks</i>	Method	CAIS			ECDT			CPB1.0			CN05		
		S-F1	P-F1	R-F1	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1
Topic drift	- w/o all	90.14	83.08	88.32	85.67	82.19	90.14	87.21	84.86	93.08	90.45	88.65	94.24
	LoCoDec (Ours)	90.95	88.43	96.39	86.70	84.94	95.34	88.44	88.07	98.98	92.06	91.79	99.91
Document-level noise	- w/o all	90.14	80.32	83.99	85.67	73.70	79.75	87.21	86.43	95.40	90.45	90.71	97.38
	LoCoDec (Ours)	90.95	90.47	99.74	86.70	87.00	99.10	88.44	88.57	99.49	92.06	91.91	99.24
Post-context noise	- w/o all	90.14	82.40	85.52	85.67	78.61	88.44	87.21	85.81	93.83	90.45	88.53	94.53
	LoCoDec (Ours)	90.95	85.53	92.22	86.70	84.90	97.16	88.44	88.36	99.08	92.06	91.66	99.87
Pre-context noise	- w/o all	90.14	77.79	81.91	85.67	70.43	75.91	87.21	59.86	61.71	90.45	67.59	70.26
	LoCoDec (Ours)	90.95	87.43	95.15	86.70	87.54	98.40	88.44	83.90	94.42	92.06	90.13	97.70
Bidirectional context noise	- w/o all	90.14	76.95	79.02	85.67	75.58	83.18	87.21	56.85	58.68	90.45	64.79	67.54
	LoCoDec (Ours)	90.95	84.24	89.57	86.70	85.71	98.59	88.44	89.00	99.92	92.06	89.88	97.32

<i>POS / NNER Tasks</i>	Method	CTB5			CTB6			GENIA			ACE2005		
		S-F1	P-F1	R-F1	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1	S-F1	P-F1	R-F1
Topic drift	- w/o all	94.59	94.52	99.30	92.35	92.33	99.02	79.13	79.05	97.91	86.81	86.20	95.09
	LoCoDec (Ours)	95.77	95.71	99.84	93.81	93.80	99.91	79.61	79.65	99.63	87.96	87.89	98.80
Document-level noise	- w/o all	94.59	93.72	98.06	92.35	92.29	97.42	79.13	78.80	96.06	86.81	86.63	94.87
	LoCoDec (Ours)	95.77	95.73	99.91	93.81	93.63	99.50	79.61	79.66	98.91	87.96	88.02	97.67
Post-context noise	- w/o all	94.59	95.49	99.34	92.35	91.87	98.40	79.13	79.73	95.85	86.81	87.00	95.98
	LoCoDec (Ours)	95.77	96.06	99.91	93.81	93.89	99.89	79.61	79.60	99.77	87.96	88.20	97.88
Pre-context noise	- w/o all	94.59	91.97	95.87	92.35	92.33	96.46	79.13	79.84	90.35	86.81	86.99	91.92
	LoCoDec (Ours)	95.77	95.40	99.58	93.81	93.64	99.34	79.61	79.96	97.71	87.96	87.41	95.85
Bidirectional context noise	- w/o all	94.59	93.44	98.40	92.35	90.52	95.00	79.13	79.47	90.07	86.81	85.73	88.66
	LoCoDec (Ours)	95.77	95.85	99.50	93.81	94.02	99.20	79.61	79.59	97.57	87.96	89.66	97.38

Table 9: Robustness comparison between LoCoDec and - w/o all using the Llama3-8B backbone under diverse contextual perturbations. Here, - w/o all denotes the ablation that removes both Neighborhood Context and Constrained Token Paraphrasing. Best results are highlighted in bold.