Memory-enhanced Large Language Model for Cross-lingual Dependency Parsing via Deep Hierarchical Syntax Understanding

Jianjian Liu, Ying Li†, Zhengtao Yu, Shun Su, Shengxiang Gao, Yuxin Huang
Yunnan Provincial Key Laboratory of Artificial Intelligence, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, China
{jjliu_nj, yingli_hlt}@foxmail.com, ztyu@hotmail.com, 3155068938@qq.com
gaoshengxiang.yn@foxmail.com, huangyuxin2004@163.com

Abstract

Large language models (LLMs) demonstrate remarkable text generation and syntax parsing capabilities in high-resource languages. However, their performance notably declines in low-resource languages due to memory forgetting stemming from semantic interference across languages. To address this issue, we propose a novel deep hierarchical syntax understanding approach to improve the cross-lingual semantic memory capability of LLMs. First, we design a multi-task joint fine-tuning strategy to implicitly align linguistic knowledge between source and target languages in LLMs, which is leveraged to initially parse the target text. Second, we automatically construct the multilingual dependency label banks based on the statistical structure information from the Universal Dependencies (UD) data. Third, we obtain each label's memory strength via in-depth analysis of the initial parsing tree and its dependency label bank. Finally, memory strength is further exploited to guide LLMs to learn the linguistic commonalities from multilingual dependency label banks, thus activating the memory ability of weak labels. Experimental results on four benchmark datasets show that our method can dramatically improve the parsing accuracy of all baseline models, leading to new state-of-the-art results. Further analysis reveals that our approach can effectively enhance the weak syntactic label memory cognition of LLMs by combining the advantages of both implicit multi-task fine-tuning and explicit label bank guiding. Our code and dependency label banks are released at https://github.com/Flamelunar/memory_dep.

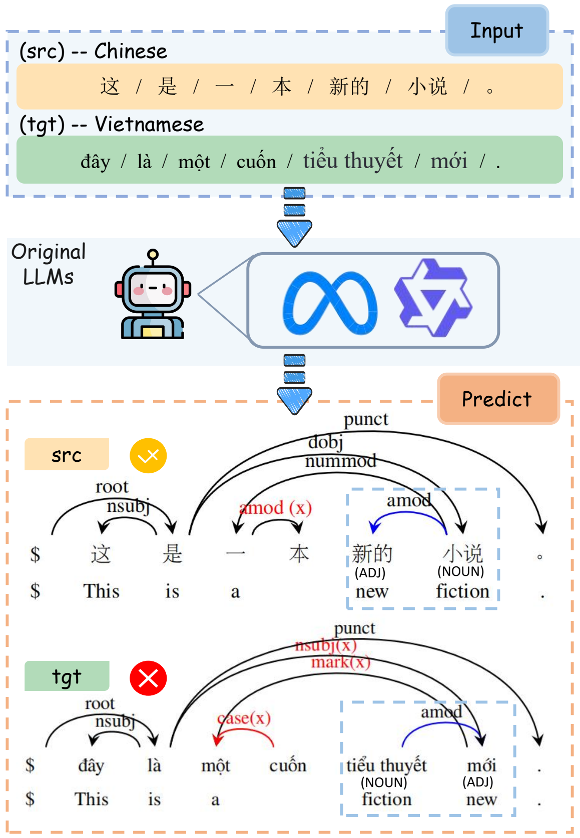

Figure 1: An example of original (unfine-tuned) LLMs dependency parsing, where high-resource source language data (Chinese) has a 85.72% correct rate and the low-resource target language data (Vietnamese) has a 57.14% correct rate. The contents of the dotted box indicate the same dependency pattern.

1 Introduction

Dependency parsing employs hierarchical tree structures to exhibit syntactic and grammatical relationships between words. As shown in Figure 1, the tree includes an arc from the headword "小说 (fiction)" to the dependent word "新的 (new)" with the label "amod", indicating adjectival modification. These hierarchical structures are widely applied in multiple natural language processing (NLP) tasks, including machine translation (Chen et al., 2023; Hou and Guo, 2024), question answering (Kang et al., 2024), grammatical error correction (Zhu et al., 2025), and text classification (Su et al., 2025). Recently, researchers

†Corresponding author.

focus on improving the syntax understanding of large language models (LLMs) using dependency trees (Chen et al., 2024a; Zhang et al., 2023; Saha and Srihari, 2024).

Advances in language models have markedly improved supervised dependency parsing for high-resource languages (Dozat and Manning, 2017; Li et al., 2019a, 2020; Ye and Teufel, 2021). However, language model-enhanced parsers are highly dependent on the scale and quality of training data, and their performances drop sharply when they are directly transferred to low-resource languages due to semantic interference (Rotman and Reichart, 2019; Wang et al., 2020; Effland and Collins, 2023). Therefore, cross-lingual dependency parsing has emerged as a promising direction, aiming to transfer effective knowledge from high-resource languages to low-resource ones (Schuster et al., 2019; Lauscher et al., 2020; Ansell et al., 2021). Existing approaches fall broadly into two categories, i.e., traditional and LLM-based methods. Traditional methods mainly rely on syntactic feature projection or transformation (He et al., 2019; Kurniawan et al., 2021; Guo et al., 2022; Choenni et al., 2023). Choudhary and O'riordan (2023) incorporate the source and target linguistic typological knowledge into a multi-task learning framework to enhance cross-lingual knowledge transfer. In contrast, LLMs (ChatGPT [1], LlaMA [2], Qwen [3], and DeepSeek [4]) exhibit remarkable generalization across a wide range of NLP tasks, benefiting from massive pre-trained corpora and highly optimized architectures. Moreover, their capabilities can be further strengthened by useful prompt learning (Zhang et al., 2024a), task-specific parameter-efficient fine-tuning (Dou et al., 2024), and retrieval augmented generation (dos Santos Junior et al., 2024).

However, LLMs struggle in low-resource languages' dependency parsing due to memory forgetting (Chen et al., 2024b; Guo et al., 2025). The main reason is that normal LLMs are prone to memorizing the semantic preferences of high-resource languages while their capability in low-resource languages is obstructed (Villalobos et al., 2024; Kuang et al., 2024). As illustrated in Figure 1, we can see that LLMs show strong parsing ability in the high-resource language (Chinese) with numerous training data, achieving a 85.72% accuracy. In contrast, the parsing accuracy of Vietnamese is only 57.14%. Concretely, although Vietnamese and Chinese share a subject–verb–object structure, they diverge in modifier placement such as Vietnamese favors post-modifiers, whereas Chinese employs pre-modifiers. Even though there is linguistic structural variation in real scenarios, the relative structure between the dependency label and POS tags is constant. For example, both Chinese and Vietnamese have a dependent word with POS tag "ADJ" modifies the head word with POS tag "NOUN", owning the same dependency label "amod". Hence, dependency relations (head–dependent patterns) often remain consistent across languages, these cross-linguistic syntactic similarities can be leveraged to improve parsing performance of low-resource languages (Hämmerl et al., 2024; Zhang et al., 2024c).

To alleviate this drawback, we propose a deep hierarchical syntax-aware approach to enhance the semantic memory capability of LLMs. First, we employ a multi-task joint fine-tuning strategy to implicitly align LLMs' syntactic knowledge across different languages. Meanwhile, fine-tuned LLMs are utilized to yield the initial parsing trees of the target language data. Then, we construct multilingual dependency label banks by extracting statistical patterns from the universal dependency treebanks. Next, each label's memory strength is estimated through structural analysis of the initial parsing trees and its distribution in the label bank. Finally, memory strength is used to guide LLMs in capturing cross-lingual syntactic commonalities, thereby reinforcing the memory capability of weak dependency labels. Experiments on four benchmark datasets demonstrate substantial performance gains in low-resource scenarios, achieving prior state-of-the-art results. Further analysis indicates that our approach can effectively strengthen the weak syntactic label memory strength of LLMs by integrating the advantages of both implicit multi-task fine-tuning and explicit dependency label bank guiding.

2 Related Work

Cross-lingual dependency parsing. Cross-lingual dependency parsing aims to transfer syntactic knowledge from high-resource to low-resource languages (Langedijk et al., 2022; Shi et al., 2022; Choenni et al., 2023). Prior work primarily re-

[1] https://openai.com/blog/chatgpt
[2] https://www.llama.com/
[3] https://tongyi.aliyun.com/
[4] https://www.deepseek.com/

lies on transfer learning to extract shared syntactic features from source languages (Eronen et al., 2023; Li et al., 2024; Liu et al., 2025). Sun et al. (2023) propose a cross-lingual self-training framework to transfer parsers from monolingual treebanks to multiple target languages. Recently, the emergence of LLMs has brought advances in causal reasoning and syntactic understanding, supporting a wide range of artificial intelligence tasks (Ma et al., 2023; Ge et al., 2024; Lin et al., 2024). Li et al. (2023) leverage LLMs in self-training by extracting grammar rules from the source domain to improve target domain parsing. Chen et al. (2024a) apply conditional mutual information to model bi-lexical dependencies, integrating grammatical constraints to strengthen unsupervised LLM-based parsing. Zhang et al. (2025) guide a lightweight LLM to generate phrase structures using grammar rules and lexical heads for data augmentation in the target domain. These studies highlight the potential of LLMs to transfer syntactic knowledge across languages. Yet two core challenges remain: incomplete learning of language-specific syntax during pretraining, and weak retention of cross-lingual patterns in LLM memory.

Syntax understanding. Syntax plays a fundamental role in natural language processing, especially in deep learning approaches (Linzen and Baroni, 2021; Aliti, 2024; Ahuja et al., 2024). Zhang et al. (2024b) leverage the "not-so-perfect" noisy syntax trees generated by unsupervised derivations and modern Chinese syntax parsers to enhance model understanding of ancient Chinese. Fan et al. (2025) propose a syntax-opinion-sentiment reasoning chain to deepen LLMs' syntax understanding for enhancing aspect-based sentiment analysis. However, most of these efforts only limit the output of the LLMs using limited knowledge to improve task-specific performance, lacking specific knowledge-infused fine-tuning for optimizing deeper parameters of the LLMs.

Memory enhancement. LLMs possess remarkable memory capacity and comprehension abilities for high-frequency information. This capability stems from their extensive parameterization and sophisticated deep neural architectures, which enable effective extraction and modeling of high-frequency data patterns during the pre-training phase (Xu et al., 2025; Zhao et al., 2024; Kim et al., 2024). Most researchers attempt to utilize or activate the deep memory of LLMs to enhance natural language processing tasks. Zhong et al. (2024) design a long-term memory mechanism to achieve LLMs' personalized interaction and long-term contextual understanding by storing, retrieving, and dynamically updating memories. Hou et al. (2024) propose a novel human-like memory architecture to enable agents to autonomously recall memories necessary for response generation, effectively addressing a limitation in the temporal cognition of LLMs, enhancing long-term dialogue capability. Inspired by the above works, we design a deep hierarchical syntax understanding method to optimize LLMs' weak syntactic label memory cognition through implicit multi-task fine-tuning and explicit dependency label bank guiding, thus improving cross-lingual dependency parsing performance.

3 Our Approach

In this work, we propose a deep hierarchical syntax understanding approach to strengthen cross-lingual semantic memory in LLMs. First, we jointly employ cross-lingual part-of-speech (POS) tagging and dependency parsing tasks to fine-tune parameters of LLMs, thus implicitly aligning linguistic knowledge between source and target languages. Meanwhile, we utilize fine-tuned LLMs to generate initial parsing trees for target language test sentences. Second, we build multilingual dependency label banks by extracting statistical syntactic patterns from universal dependency corpora, which explicitly exhibit the relationship between common dependency labels and fine-grained POS tags. Then, we analyse each label's correct rate in initial parsing trees and the distribution frequency in fine-tuning training data to identify its memory strength. Finally, memory strength is further exploited to guide LLMs to learn the linguistic commonalities from multilingual dependency label banks, yielding more accurate final parsing trees. Figure 2 shows the overall architecture with three components, i.e., *multi-task joint fine- tuning*, *dependency label bank construction*, *hierarchical memory enhancement*.

3.1 Multi-task Joint Fine-tuning

Although the LLMs have some generalization ability on most natural language processing tasks, their syntax understanding and parsing capability on low-resource languages is not activated. Hence, we propose the multi-task joint fine-tuning method, which employs cross-lingual POS tagging as an auxiliary task to activate the implicit cross-lingual

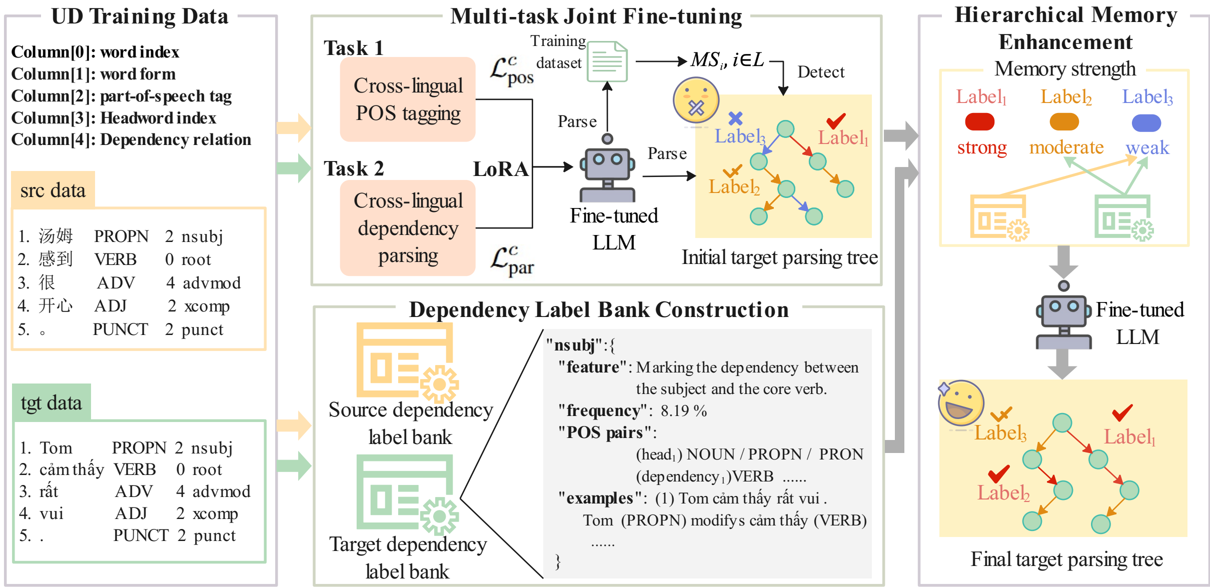

Figure 2: The overall architecture of our method.

semantic alignment capability of LLMs.

For each input sentence which contains golden language type, POS tags, and dependency trees, LLMs first convert it into high-dimensional feature vectors $\mathbf{x}$. Then, Low-Rank Adaptation (LoRA) is leveraged to fine-tune LLMs by learning pairs of rank decomposition matrices while keeping the original weights frozen (Hu et al., 2022). Formally, considering that a linear layer is defined as $\mathbf{y} = \mathbf{W}\mathbf{x}$ with the weight matrix $\mathbf{W}$. LoRA modifies it into $\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{B}\mathbf{A}\mathbf{x}$, where $\mathbf{W} \in \mathcal{R}^{d\times k}$, $\mathbf{B} \in \mathcal{R}^{d\times r}$, $\mathbf{A} \in \mathcal{R}^{r\times k}$, and $r \ll \min(d, k)$, which greatly reduces the amount of parameters needed to be learned. Meanwhile, we employ the cross-entropy loss function to train two tasks until LLMs converge or reach the maximum number of training epochs. The formulas of cross-lingual POS tagging loss $\mathcal{L}^c_{\text{pos}}$ and cross-lingual dependency parsing loss $\mathcal{L}^c_{\text{par}}$ are computed as follows,

$$\mathcal{L}^c_{\text{pos}} = -\sum_{i=1}^{P} p_i \log(\hat{p}_i) - \sum_{k=1}^{T} t_k \log(\hat{t}_k) \quad (1)$$

$$\begin{aligned}\mathcal{L}^c_{\text{par}} = &-\sum_{i=1}^{H} h_i \log(\hat{h}_i) - \sum_{j=1}^{L} l_j \log(\hat{l}_j) \\ &-\sum_{k=1}^{T} t_k \log(\hat{t}_k)\end{aligned} \quad (2)$$

where P, H, L, and T are the number of POS tags, headwords, dependency labels, and language types, respectively. p_i, h_i, l_j, and t_k represent the gold-standard POS tags, headwords, dependency labels and language types distribution probability, that only one element is 1 corresponding to the correct index. Finally, the parameters of the LLMs are optimized by minimizing the total loss $\mathcal{L}$.

$$\mathcal{L} = \mathcal{L}^c_{\text{pos}} + \mathcal{L}^c_{\text{par}} \quad (3)$$

After obtaining the best fine-tuned LLMs, we utilize them to parse the target language sentences and yield initial parsing trees Y^{ini}.

3.2 Dependency Label Bank Construction

Fine-tuned LLMs exhibit improved dependency parsing capabilities in low-resource languages. However, some dependency labels appear too rarely in training data, limiting the LLMs' syntactic comprehension and memory retention of these structures. To address this, we construct two dependency label banks based on the universal dependency training datasets of the source and target languages. Each dependency label bank explicitly exhibits the relationship between common dependency labels and fine-grained POS tags. As shown in Figure 2, each dependency label object includes four keys, i.e., *feature*, *frequency*, *POS pairs*, and *examples*.

Concretely, we first employ fine-tuned LLMs to summarize the characteristics, usage, and meaning as its *feature* value. Next, we compute the percentage of each dependency label distribution across the total number in the fine-tuned training data as its *frequency* value. This frequency metric reflects the memory strength of LLMs for each label. For each label, we then extract head–dependent word pairs to generate part-of-speech (POS) combinations and record the frequency of each POS pair as the value of *POS pairs*. Finally, we select three representative sentences with their explanation for each POS pair from the corpus to serve as the *examples* attribute.

3.3 Hierarchical Memory Enhancement

To identify weak memory dependency labels in LLMs, we first compute a memory strength score $MS_i \in [0, 1]$ for each dependency label. This memory strength score is based on the correct rate $c_i \in [0, 1]$ of each dependency label and the frequency $f_i \in [0, 1]$ of dependent labels in the fine-tuned training data. Concretely, in the training process, we use the golden labels of the training dataset to fine-tune the LLM. Then, the fine-tuned LLM is used to parse the training sentences. Next, we count the correct rate c_i of each dependency label based on LLM predicted and golden labels in the training dataset. In the test process, considering that the distributions of dependency labels in the training and test datasets are highly similar, we use the yielded correct rate c_i in the training process to calculate the label memory strength. Inspired by the memory forgetting formula of Zhong et al. (2024), our improved memory strength formula is calculated as follows,

$$MS_i\left(c_i, f_i\right) = c_i\left(1 - e^{-\lambda f_i}\right) \tag{4}$$

where the memory factor $\lambda \in [0, 100]$ controls the relative influence of frequency and correct rate. The larger value increases the impact of f_i, while the smaller value emphasises the impact of c_i. Then, we enhance syntax memory hierarchically based on three categorized memory strength tiers.

As shown in Algorithm 1, labels with $MS_i < 0.6$ are considered weak memories, which are augmented using knowledge from both source and target language dependency label banks. Labels with $0.6 \leq MS_i < 0.9$ are moderate memory, which are refined using target language data alone. Labels with $MS_i \geq 0.9$ are strong memory, which does not require further augmentation. Finally, the initial parsing trees Y^{ini} are corrected by memory enhancement, thus obtaining more accurate final parsing trees Y^{fin}.

Algorithm 1: Hierarchical Memory Enhancement

Input: L from initial parsing trees Y^{ini}, each dependency label's correct rate c_i and frequency f_i, source dependency label bank D^s and target dependency label bank D^t.
Hyperparameters: Impact factor λ

```
1: For L_i ∈ L:
2:    MS_i(c_i, f_i) = c_i(1 − e^{−λ f_i})
3:    if MS_i < 0.6:
4:       Y^fin ← L_i + D^s + D^t
5:    elif 0.6 ≤ MS_i < 0.9:
6:       Y^fin ← L_i + D^t
7:    else:
8:       Y^fin ← L_i
```

Table 1: Hierarchical memory enhancement.

Dataset	Train	Dev	Test	All
		UD public datasets		
English (EWT)	12,544	2,001	2,077	16,622
Chinese(GSDSimp)	3,997	500	500	4,997
Vietnamese (VTB)	1,400	1,123	800	3,323
Tamil (TTB)	400	80	120	600
Coptic (Scriptorium)	1,419	381	403	2,203
Maltese (MUDT)	1,123	433	518	2,074

Table 2: Dataset statistics in sentence number.

4 Experiments

4.1 Experimental Setups

Datasets. We acquiescently experimented with using Chinese (zh) as the source language for Vietnamese (vi) and Tamil (ta) while English (en) is the source language for Coptic (cop), and Maltese (mt), which are all derived from the Universal Dependencies (UD) v2.13 treebank [5]. Moreover, we use all languages' training datasets to fine-tune large language models (LLMs) and evaluate on their respective test datasets. Detailed dataset statistics are presented in Table 2.

Evaluation. We utilize Labeled Attachment Score (LAS) and Unlabeled Attachment Score (UAS) as evaluation metrics (Liu et al., 2025). All models are trained for no more than 1000 iterations, and their performances are evaluated on the development dataset after each iteration to guide the model selection.

Hyperparameter choices. 1) Training traditional parsers. We set the parameters of the three traditional small models uniformly according to the most hyperparameter settings of Li et al. (2019a), including MLP and BiAffine dimensions and learning rates. 2) Fine-tuning large language models. The key hyperparameters are set as in Table 3, the

[5] https://universaldependencies.org/

Hyperparameter	Value	
	LoRA	*QLoRA (8-bit)*
lora_alpha	16	8
lora_rank	8	4
loraplus_lr_ratio	16	8
num_train_epochs	5	5
compute_type	bf16	bf16
leariining_rate	5e-5	5e-5
cutoff_len	3500	3500

Table 3: Hyperparameter setting of fine-tuning LLMs.

rest of the hyperparameters take on default values.

Baselines. We employ three typical cross-lingual models and three large language models as baseline models to demonstrate the effectiveness of our approach.

1) Three typical cross-lingual models. During the training process of three typical cross-lingual models, we use source and target language training datasets to train models and evaluate its performance on target language test dataset. ***Full Shared Model (FulSha).*** Peng et al. (2017) enhance heterogeneous dependency parsing by employing fully shared encoder parameters across three dependency graph formalisms to capture cross-formalism commonalities. Following a similar strategy, we share all model parameters and alternately train the BiAffine parser (Dozat and Manning, 2017) on both source and target language datasets. ***Language Embedding Model (LanEmb).*** Li et al. (2019b) show that injecting domain embeddings as auxiliary inputs improves cross-domain parsing by informing the model of domain-specific characteristics. Analogously, we introduce 8-dimensional language embeddings to explicitly encode language identity, guiding the model in distinguishing between different language structures. ***Multi-task Learning Model (MulLea).*** Building on Dou et al. (2023), who leverage named entity recognition (NER) as an auxiliary task to transfer lexical knowledge across domains, we treat source language parsing as an auxiliary task to facilitate syntactic knowledge transfer to the target low-resource language. ***w/ roberta.*** For all typical models above, we use the XLM-RoBERTa-base [6] pre-training model to extract the corresponding feature representations of the input words and add them to the random word embeddings of the above models to enhance the contextual representation of the words.

2) Three large language models. To validate the effectiveness of our approach, we set zero-shot, one-shot, and Five-shot for three large language models. Due to the original LLM's poor parsing performance (or incorrect parsing formatting) for low-resource languages and ensuring cross-lingual evaluation, we first translate target language texts into the source language (Chinese or English) and parse them using pre-trained BiAffine parsers. Then, resulting syntactic trees are added to prompts for structural references, enabling the cross-lingual settings. ***Llama3.1-8B-Instruct.*** Which is Meta's lightweight open-source model, featuring a 128k-token context window. It excels in English-centric tasks, including instruction following and code generation, making it suitable for applications requiring deep contextual understanding. ***Qwen2.5-7B-Instruct.*** Which is a 7B parameter instruction-tuned model optimized for multilingual tasks, particularly strong in East and Southeast Asian languages such as Chinese, Vietnamese, and Korean. It demonstrates robust performance in mathematical reasoning and code generation within multilingual contexts. ***Qwen2.5-14B-Instruct.*** Which is a 14.7B parameter model with a 128 K-token context window. It excels in processing structured data (e.g., tables, JSON) and generating long-form content, making it ideal for applications involving complex documents and multilingual content.

4.2 Main Results

Table 4 presents the main results of baseline models and our method across three LLMs. We first evaluate three LLMs under zero-shot, one-shot, and few-shot settings for cross-lingual dependency parsing. As expected, performance improves with more examples in prompt learning. The Qwen series outperforms others, and its performance scales with model size. Next, our implicit multi-task joint training strategy can enhance parsing accuracy dramatically. Then, LLMs' performance is further boosted by applying our explicit dependency label bank to correct weak-memory syntactic patterns, demonstrating our method's effectiveness. Finally, we find that LLMs with more parameters perform better when using our approach. For instance, our method on "Qwen2.5-14B-Instruct" surpasses all baselines of traditional models and LLMs, proving considerable room for further improvement.

We compare our models with several previous works on traditional models. Kondratyuk and Straka (2019) propose UDify, a multilingual BERT-

[6] https://huggingface.co/xlm-roberta-base

Model	Vietnamese		Tamil		Coptic		Maltese		Avg.	
	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS
Results of previous works										
UDify(2019)	66.00	74.11	68.29	78.34	10.82	27.58	75.56	83.07	55.17	65.78
MBERT(2022)	61.24	70.45	54.94	62.35	82.11	86.87	72.69	80.54	67.75	75.05
ESR(2023)	60.80	70.21	66.40	74.12	77.34	81.42	74.20	82.34	69.69	77.02
Dynamic(2025)	66.75	80.03	69.18	79.09	86.32	89.95	76.19	83.28	74.61	83.09
Compare with traditional models										
FulSha	54.82	69.02	56.79	66.76	72.28	76.60	68.42	76.61	63.08	72.25
MulLea	56.21	70.01	57.02	67.54	73.52	77.41	67.24	75.14	63.50	72.53
LanEmb	55.89	70.09	57.27	69.28	72.04	76.42	69.01	77.35	63.55	73.29
FulSha (w/ roberta)	62.53	78.94	63.15	77.23	79.28	85.60	72.79	81.61	69.44	80.85
MulLea (w/ roberta)	64.37	79.26	63.90	75.82	82.59	87.41	70.15	79.75	70.25	80.56
LanEmb (w/ roberta)	63.52	79.28	64.25	78.18	79.14	85.52	73.01	81.74	70.23	81.18
Compare with large language models										
Llama3.1-8B-Instruct										
Zero-shot	15.57	30.03	9.45	22.12	9.59	19.82	17.49	38.08	13.03	27.76
One-shot	18.93	34.65	15.65	30.07	11.87	23.84	19.59	41.41	16.51	32.49
Five-shot	21.79	36.80	18.68	32.65	12.82	25.90	30.21	44.06	20.88	34.85
LoRA	56.66	69.33	57.45	68.07	69.02	74.03	70.44	75.97	63.39	71.85
Our	**60.12**	**72.45**	**61.05**	**71.52**	**73.65**	**77.34**	**75.14**	**78.13**	**67.49**	**74.86**
Qwen2.5-7B-Instruct										
Zero-shot	18.29	33.87	14.95	34.92	9.37	22.16	19.30	38.97	15.48	32.48
One-shot	20.23	37.03	17.90	35.68	9.72	23.23	20.12	42.06	16.99	34.50
Five-shot	23.08	38.02	23.00	37.30	11.85	25.60	28.18	43.96	21.53	36.22
LoRA	63.26	76.27	55.97	67.68	75.65	80.20	70.22	76.64	66.28	75.20
Our	**66.48**	**79.35**	**60.42**	**70.54**	**79.42**	**83.14**	**74.31**	**79.62**	**70.16**	**78.16**
Qwen2.5-14B-Instruct										
Zero-shot	24.85	41.71	23.68	38.50	13.84	27.89	31.25	49.15	23.41	39.31
One-shot	26.50	43.45	25.15	39.32	15.12	29.03	33.42	51.32	25.05	40.78
Five-shot	28.46	45.96	29.23	43.61	17.35	31.47	36.54	53.75	27.90	43.70
QLoRA	66.24	79.93	63.45	74.48	83.10	86.79	76.23	82.28	72.26	80.87
Our	**68.51**†	**83.14**†	**65.57**†	**77.63**†	**86.42**†	**90.02**†	**78.39**†	**85.31**†	**74.72**†	**84.03**†

Table 4: Main results of four languages on the test dataset. "w/ roberta" represents the enhancement of word vectors via XLM-RoBERTa-base pre-trained model at the input layer. † indicates the best performance across all methods. All experiments are conducted on GeForce RTX 3090 24GB GPUs, using up to 2 GPUs for LoRA or QLoRA.

based model fine-tuned across 104 languages for enhanced parsing. Moreover, Gessler and Zeldes (2022) employ a vocabulary expansion method and fine-tune BERT to enhance parsing performance. Lastly, Effland and Collins (2023) apply expected statistic regularization with low-order multi-task structural features to refine distributions. Liu et al. (2025) propose dynamic syntactic networks that filter harmful source-language features while amplifying cross-lingual syntactic commonalities. In contrast, our approach jointly fine-tunes LLMs for deep syntactic understanding and uses the dependency label bank to strengthen weak syntactic memory, outperforming previous methods. These results confirm the efficacy and potential of our approach.

4.3 Ablation Study

Table 5 presents a detailed ablation analysis on both the LoRA fine-tuning process and the dependency label bank usage. For the LoRA process, removing the cross-lingual POS tagging task leads to a performance drop, indicating that POS information supports syntactic learning in LLMs. Then, eliminating the source language dependency parsing task causes an even larger decline, suggesting it contributes essential syntactic knowledge for understanding the target language. When both tasks are removed, performance degrades most severely, indicating their complementary value. For the dependency label bank usage, omitting both the source and target language dependency label banks reduces performance. Then, we find that enhancing knowledge directly from the target language proves more effective. In addition, completely removing all dependency banks causes further degradation, confirming their overall utility.

4.4 Error Analysis

Sentence Lengths Figure 3 reports LAS across sentence lengths. First, Parsing accuracy declines significantly beyond 30 words, with an av-

Model	Llama3.1-8B		Qwen2.5-14B	
	LAS	UAS	LAS	UAS
	LoRA & QLoRA ablation study			
Our	**59.68**	**72.12**	**68.07**	**82.41**
w/o pos	56.13	68.24	63.45	76.27
w/o src_dp	49.21	61.47	54.24	76.51
w/o src_dp & pos	47.32	59.17	52.70	74.28
	Dependency label banks ablation study			
Our	**59.68**	**72.12**	**68.07**	**82.41**
w/o src	58.52	71.13	67.54	81.67
w/o tgt	57.13	69.74	66.37	80.57
w/o src & tgt	56.34	68.93	65.67	79.76

Table 5: The ablation study on the Vietnamese development dataset. "w/o pos" means removing the cross-lingual POS tagging task. "w/o src_dp" means removing the source language dependency parsing task. "w/o src" or "w/o tgt" means not using the dependency label bank of the source or target language.

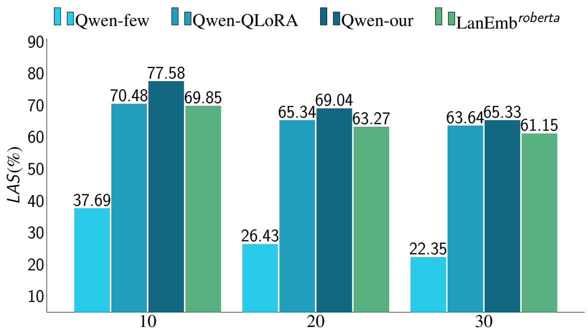

Figure 3: LAS for various sentence lengths on the Vietnamese development dataset, where "Qwen" is Qwen2.5-14B-Instruct.

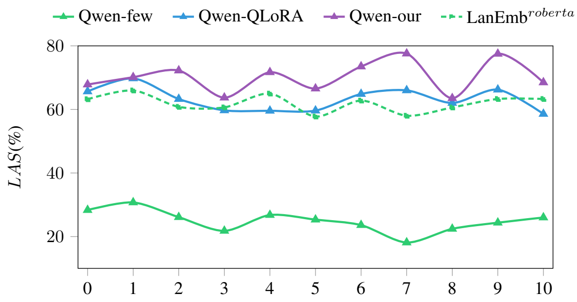

Figure 4: LAS curves regarding dependency distances on the Vietnamese development dataset, where "Qwen" is Qwen2.5-14B-Instruct.

DEP	Accuracy (%)			
	Qwen2.5-14B-Instruct			LanEmb
	few-shot	QLoRA	our	w/ roberta
advmod	49.07	86.47	**87.12**	77.19
amod	48.35	59.69	**64.11**	61.24
case	62.63	83.14	**85.33**	75.50
cc	84.00	64.16	**74.24**	78.85
ccomp	12.86	48.03	**61.11**	43.61
conj	58.27	75.47	**80.00**	71.03
det	33.14	95.73	**97.00**	78.14
mark	41.18	83.64	**84.71**	73.88
nmod	21.44	62.25	**67.92**	50.99
nsubj	68.24	86.93	**88.49**	83.75
obl	14.08	38.32	**50.41**	32.31
root	48.27	81.14	**84.16**	77.83
xcomp	23.60	49.33	**57.63**	44.77

Table 6: Dependency label accuracy on the Vietnamese development dataset.

erage drop of 10.78 points, exhibiting the difficulty of long-sentence parsing. The "Qwen-few" model consistently underperforms, reflecting the limited parsing ability of standard LLMs in low-resource languages. However, multi-task joint fine-tuning "Qwen-QLoRA" markedly enhances performance. Moreover, incorporating our dependency label bank further boosts performance, suggesting that source-language syntactic patterns enhance the LLMs' syntax understanding of the target language. Overall, our approach outperforms the benchmark "LanEmb$^{\text{roberta}}$", affirming its effectiveness.

Dependency Distances Figure 4 presents LAS about absolute dependency distances. First, the "Qwen-few" model consistently underperforms across most distances. In contrast, the "Qwen-QLoRA" model significantly improves dependency parsing accuracy for both short and long distances. Then the "Qwen-our" model achieves the highest performance, surpassing "LanEmb$^{\text{roberta}}$", demonstrating that our multi-task joint fine-tuning and dependency label bank can enhance dependency parsing capabilities at all distances via learn syntax commonalities across languages.

Dependency labels. Table 6 reports accuracy scores for dependency label predictions. First, "QLoRA" outperforms the "few-shot" baseline, suggesting that multi-task joint fine-tuning enables better cross-lingual syntactic generalization. Then, accuracy improves further with the addition of our dependency label bank, surpassing the "LanEmb$^{\text{roberta}}$" model across most labels. These findings highlight the effectiveness of combining implicit fine-tuning with explicit memory enhancement to optimize parsing in low-resource languages.

5 Conclusion

We propose a novel deep hierarchical syntax understanding method to enhance the weak dependency label memory capability in large language models. Concretely, we exploit implicit multi-task fine-tuning and explicit dependency label bank guiding to boost LLMs to absorb cross-lingual syntactic commonalities. Experiments on four benchmark datasets show substantial accuracy gains across all baseline models, achieving state-of-the-art performance. Analysis reveals that both multi-task joint fine-tuning and extra dependency label bank can extract useful syntactic knowledge from the source language to enhance the target language parsing accuracy. Moreover, in-depth comparison demonstrates that our method can alleviate semantic interference across languages and improve the memory strength of most dependency labels, thus further improving the parsing performance.

Limitations

The large language models used in our experiments was not sufficient to cover most of them, while we did not try to include more useful auxiliary knowledge inside the dependency bank, which we will continue to delve into in our future work.

Ethical Considerations

Competing interests All authors declare no competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. **Ethics approval and consent to participate** This article does not contain any studies with human participants performed by any of the authors. **Data availability** The data used in this study are from publicly available datasets. The Universal Dependencies (UD) datasets used in this study are publicly available and can be accessed through `https://universaldependencies.org/`. **Code availability** The code and bilingual dictionary used to support this work can be accessible through `https://github.com/Flamelunar/memory_dep`.

Acknowledgements

We thank all reviewers for their helpful comments. This work is financially supported by the National Natural Science Foundation of China (62306129, U21B2027, 62366027, 62266028), Yunnan Fundamental Research Projects (202401CF070121, 202401BC070021, 202301AS070047), Yunnan Provincial Major Science and Technology Special Plan Projects (202502AD080016, 202402AG050007, 202103AA080015, 202202AD080003, 202203AA080004), Kunming University of Science and Technology "Double First-rate" Construction Joint Project (202301BE070001-027, 202201BE070001-021), Yunnan High and New Technology Industry Project (201606).

References

Kabir Ahuja, Vidhisha Balachandran, Madhur Panwar, Tianxing He, Noah A Smith, Navin Goyal, and Yulia Tsvetkov. 2024. Learning syntax without planting trees: Understanding when and why transformers generalize hierarchically. In *ICML 2024 Workshop on Mechanistic Interpretability*.

Afrim Aliti. 2024. Exploring the role of syntax in language comprehension and production. *International Scientific Journal Monte (ISJM)*, 9(2).

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. Mad-g: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of EMNLP*, pages 4762–4781.

Junjie Chen, Xiangheng He, and Yusuke Miyao. 2024a. Language model based unsupervised dependency parsing with conditional mutual information and grammatical constraints. In *Proceedings of NAACL-HLT*, pages 6355–6366, Mexico City, Mexico.

Junjie Chen, Xiangheng He, and Yusuke Miyao. 2024b. Language model based unsupervised dependency parsing with conditional mutual information and grammatical constraints. In *Proceedings of NAACL*, pages 6355–6366.

Xinran Chen, Yuran Zhao, Jianming Guo, Sufeng Duan, and Gongshen Liu. 2023. Sdpsat: Syntactic dependency parsing structure-guided semi-autoregressive machine translation. In *International Conference on Neural Information Processing*, pages 604–616. Springer.

Rochelle Choenni, Dan Garrette, and Ekaterina Shutova. 2023. Cross-lingual transfer with language-specific subnetworks for low-resource dependency parsing. *Computational Linguistics*, 49(3):613–641.

Chinmay Choudhary and Colm O'riordan. 2023. Multilingual end-to-end dependency parsing with linguistic typology knowledge. In *proceedings of SIGTYP*, pages 12–21.

José Cassio dos Santos Junior, Rachel Hu, Richard Song, and Yunfei Bai. 2024. Domain-driven llm development: Insights into rag and fine-tuning practices. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6416–6417.

Chenxiao Dou, Xianghui Sun, Yaoshu Wang, Yunjie Ji, Baochang Ma, and Xiangang Li. 2023. Domain-adapted dependency parsing for cross-domain named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12737–12744.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, et al. 2024. Loramoe: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *Proceedings of ACL*, pages 1932–1945.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of ICLR*.

Thomas Effland and Michael Collins. 2023. Improving low-resource cross-lingual parsing with expected statistic regularization. *TACL*, pages 122–138.

Juuso Eronen, Michal Ptaszynski, and Fumito Masui. 2023. Zero-shot cross-lingual transfer language selection using linguistic similarity. *Information Processing & Management*, 60(3):103250.

Rui Fan, Shu Li, Tingting He, and Yu Liu. 2025. Aspect-based sentiment analysis with syntax-opinion-sentiment reasoning chain. In *Proceedings of COLING*, pages 3123–3137, Abu Dhabi, UAE.

Yingqiang Ge, Wenyue Hua, Kai Mei, Juntao Tan, Shuyuan Xu, Zelong Li, Yongfeng Zhang, et al. 2024. Openagi: When llm meets domain experts. *Advances in Neural Information Processing Systems*, 36.

Luke Gessler and Amir Zeldes. 2022. MicroBERT: Effective training of low-resource monolingual BERTs through parameter reduction and multitask learning. In *Proceedings of ACL-MRL*, pages 86–99.

Peiming Guo, Shen Huang, Peijie Jiang, Yueheng Sun, Meishan Zhang, and Min Zhang. 2022. Curriculum-style fine-grained adaption for unsupervised cross-lingual dependency transfer. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:322–332.

Shuaishuai Guo, Yanhu Wang, Jia Ye, Anbang Zhang, Peng Zhang, and Kun Xu. 2025. Semantic importance-aware communications with semantic correction using large language models. *IEEE Transactions on Machine Learning in Communications and Networking*.

Katharina Hämmerl, Jindřich Libovickỳ, and Alexander Fraser. 2024. Understanding cross-lingual alignment—a survey. In *Findings of ACL*, pages 10922–10943.

Junxian He, Zhisong Zhang, Taylor Berg-Kirkpatrick, and Graham Neubig. 2019. Cross-lingual syntactic transfer through unsupervised adaptation of invertible projections. In *Proceedings of ACL*, pages 3211–3223.

Yuki Hou, Haruki Tamoto, and Homei Miyashita. 2024. " my agent understands me better": Integrating dynamic human-like memory recall and consolidation in llm-based agents. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Zhenyu Hou and Junjun Guo. 2024. Virtual visual-guided domain-shadow fusion via modal exchanging for domain-specific multi-modal neural machine translation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 4227–4235.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Long Kang, Xiaoge Li, and Xiaochun An. 2024. Knowledge-aware adaptive graph network for commonsense question answering. *Journal of Intelligent Information Systems*, pages 1–20.

Byeongho Kim, Sanghoon Cha, Sangsoo Park, Jieun Lee, Sukhan Lee, Shin-haeng Kang, Jinin So, Kyungsoo Kim, Jin Jung, Jong-Geon Lee, et al. 2024. The breakthrough memory solutions for improved performance on llm inference. *IEEE Micro*, 44(3):40–48.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of EMNLP-IJCNLP*, pages 2779–2795.

Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. 2024. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5260–5271.

Kemal Kurniawan, Lea Frermann, Philip Schulz, and Trevor Cohn. 2021. PPT: Parsimonious parser transfer for unsupervised cross-lingual adaptation. In *Proceedings of ACL*, pages 2907–2918, Online.

Anna Langedijk, Verna Dankers, Phillip Lippe, Sander Bos, Bryan Cardenas Guevara, Helen Yannakoudakis, and Ekaterina Shutova. 2022. Meta-learning for fast cross-lingual adaptation in dependency parsing. In *Proceedings of ACL*, pages 8503–8520.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual transformers. In *Proceedings of EMNLP*, pages 4483–4499.

Jianling Li, Meishan Zhang, Peiming Guo, Min Zhang, and Yue Zhang. 2023. Llm-enhanced self-training for cross-domain constituency parsing. In *Proceedings of EMNLP*, pages 8174–8185.

Ying Li, Zhenghua Li, Min Zhang, Rui Wang, Sheng Li, and Luo Si. 2019a. Self-attentive biaffine dependency parsing. In *Proceedings of IJCAI*, pages 5067–5073.

Ying Li, Jianjian Liu, Zhengtao Yu, Shengxiang Gao, Yuxin Huang, and Cunli Mao. 2024. Representation alignment and adversarial networks for cross-lingual dependency parsing. In *Findings of EMNLP*, pages 7687–7697, Miami, Florida, USA.

Zhenghua Li, Xue Peng, Min Zhang, Rui Wang, and Luo Si. 2019b. Semi-supervised domain adaptation for dependency parsing. In *Proceedings of ACL*, pages 2386–2395.

Zuchao Li, Hai Zhao, and Kevin Parnow. 2020. Global greedy dependency parsing. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8319–8326.

Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. 2024. Data-efficient fine-tuning for llm-based recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 365–374.

Tal Linzen and Marco Baroni. 2021. Syntactic structure from deep learning. *Annual Review of Linguistics*, 7(1):195–212.

Jianjian Liu, Zhengtao Yu, Ying Li, Yuxin Huang, and Shengxiang Gao. 2025. Dynamic syntactic feature filtering and injecting networks for cross-lingual dependency parsing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24614–24622.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Hao Peng, Sam Thomson, and Noah A Smith. 2017. Deep multitask learning for semantic dependency parsing. In *Proceedings of ACL*, pages 2037–2048.

Guy Rotman and Roi Reichart. 2019. Deep contextualized self-training for low resource dependency parsing. *TACL*, 7:695–713.

Sougata Saha and Rohini K Srihari. 2024. Turiya at dialam-2024: Inference anchoring theory based llm parsers. In *Proceedings of the 11th Workshop on Argument Mining (ArgMining 2024)*, pages 124–129.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of NAACL-HLT*, pages 3795–3805.

Freda Shi, Kevin Gimpel, and Karen Livescu. 2022. Substructure distribution projection for zero-shot cross-lingual dependency parsing. In *Proceedings of ACL*, pages 6547–6563.

Shun Su, Dangguo Shao, Lei Ma, Sanli Yi, and Ziwei Yang. 2025. Adcl: An attention feature enhancement network based on adversarial contrastive learning for short text classification. *Advanced Engineering Informatics*, 65:103202.

Kailai Sun, Zuchao Li, and Hai Zhao. 2023. Cross-lingual universal dependency parsing only from one monolingual treebank. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Position: Will we run out of data? limits of llm scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020. Extending multilingual BERT to low-resource languages. In *Findings of EMNLP*, pages 2649–2656.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*.

Yuxiao Ye and Simone Teufel. 2021. End-to-end argument mining as biaffine dependency parsing. In *Proceedings of EACL*, pages 669–678.

Collin Zhang, John Morris, and Vitaly Shmatikov. 2024a. Extracting prompts by inverting llm outputs. In *Proceedings of EMNLP*, pages 14753–14777.

Meishan Zhang, Peiming Guo, Min Zhang, Yue Zhang, et al. 2023. Llm-enhanced self-training for cross-domain constituency parsing. In *Proceedings of EMNLP*, pages 8174–8185.

Shitou Zhang, Ping Wang, Zuchao Li, Jingrui Hou, and Qibiao Hu. 2024b. Confidence-based syntax encoding network for better ancient chinese understanding. *Information Processing & Management*, 61(3):103616.

Yuhong Zhang, Jianqing Wu, Kui Yu, and Xindong Wu. 2024c. Diverse structure-aware relation representation in cross-lingual entity alignment. *ACM Transactions on Knowledge Discovery from Data*, 18(4):1–23.

Ziyan Zhang, Yang Hou, Chen Gong, and Zhenghua Li. 2025. Data augmentation for cross-domain parsing via lightweight LLM generation and tree hybridization. In *Proceedings of COLING*, pages 11235–11247, Abu Dhabi, UAE.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. 2024. Galore: memory-efficient llm training by gradient low-rank projection. In *Proceedings of ICML*, pages 61121–61143.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

Shichang Zhu, Jianjian Liu, Ying Li, and Zhengtao Yu. 2025. Automatical sampling with heterogeneous corpora for grammatical error correction. *Complex & Intelligent Systems*, 11(1):25.

A Effect of Memory Strength

DEP	F(%)	C(%)	MS	Bank	C^f(%)
punct	14.57	99.82	1.00	-	99.82
nsubj	8.19	86.93	0.86	t	88.49
root	6.84	81.14	0.80	t	84.16
advmod	6.43	86.47	0.85	t	87.12
case	4.83	83.14	0.79	t	85.33
conj	4.00	75.47	0.69	t	80.00
nmod	3.94	62.25	0.56	s, t	67.92
xcomp	2.81	49.33	0.40	s, t	57.63
mark	2.51	83.64	0.65	t	84.71
obl	2.22	38.32	0.28	s, t	50.41
nummod	2.04	88.57	0.62	t	91.30
amod	2.01	59.69	0.42	s, t	64.11
cc	1.92	64.16	0.44	s, t	74.24
advcl	1.80	56.21	0.37	s, t	67.16
obl:tmod	1.77	69.86	0.46	s, t	77.27
det	1.62	96.73	0.60	t	97.00

Table 7: Memory strength of some dependency labels, where the memory formula's impact factor λ is set to 60. F, C, MS, and Bank are the frequency, correct rate, memory strength, and the use of dependency label bank. "-" means no use of dependency label bank, and "s or t" means the use of the source language or the target language. C^f is the correct rate of optimized final parsing results.

Table 7 presents the memory strength of most dependency labels and the effect of using dependency label banks. We find that very few labels reach the maximum memory strength of 1, only the label "punct" because its high frequency in the fine-tuning data gives the LLMs a strong understanding of it. Then, using both source and target language dependency label banks provides a larger improvement for labels with weak memory and low initial accuracy, while using only the target language dependency label bank yields a moderate gain for labels with moderate memory strength. This suggests that sharing syntactic structures from the source language helps the LLMs better understand the target language syntax, demonstrating the validity of our method.

B Effect of Frequency and Correct Rate

Table 8 shows the influence of frequency and correct rate on memory enhancement. We find that lowering λ, which increases the weight of the LLMs' initial label correct rate when calculating memory strength, leads to improved scores. This is because it lowers the calculated memory strengths overall, causing most labels to be treated as weak memories. As a result, more information from the dependency label bank is used, but it increases the number of occupied tokens and slows down inference. In contrast, increasing λ reduces memory usage and speeds up inference but leads to lower performance. The parameters we selected strike a balance between these trade-offs and result in strong overall performance.

λ	Tokens	Time	Qwen2.5-14B-Instruct	
			LAS	UAS
30	3.5k	24s	68.24	82.97
60	2.0k	15s	68.07	82.41
90	1.5k	10s	66.32	80.24

Table 8: Impact of frequency and correct rate for memory enhancement, where increasing λ amplifies the importance of frequency and conversely emphasises the importance of correct rate.

Thresholds		Qwen2.5-14B-Instruct	
$w \to m$	$m \to s$	LAS	UAS
0.6	0.9	68.07	82.41
0.4	0.9	67.67	82.04
0.8	0.9	68.34	82.77
0.6	0.7	67.87	82.13
0.6	1.0	68.20	82.24

Table 9: Thresholds for the division of memory strength, where "$w \to m$" is the threshold that determines weak to moderate memory, "$m \to s$" is the threshold that determines moderate to strong memory.

C Influence of Different Memory Strength Thresholds

Table 9 shows the effect of different thresholds for dividing memory strength levels. The first row presents our default parameter settings. We observe that lowering the threshold between weak and moderate memory (second row) and between moderate and strong memory (fourth row) leads to a drop in performance. This happens because less knowledge from the dependency label banks is used, which reduces the benefit from syntactic structure transfer and weakens performance. In contrast, the parameter settings in the third and fifth rows expand the range of labels considered as weak or moderate memory, which increases the use of the dependency label banks and results in a slight performance gain. These results confirm the value of extracting shared syntactic structures from our memory resource.

D Fine-tuning Data Template

Table 10 and 11 illustrate the fine-tuning data templates employed in the cross-lingual POS tagging task and cross-lingual dependency parsing task. This information is mainly used to clearly show the data format used to fine-tune large language models, and the data will be publicly available in JSON format.

Instrcuct:	You are an expert in POS tagging, learn the source language sentence's part-of-speech and identify the target sentence's language type and tag part-of-speech for its tokens.
Input:	(src) 汤姆\t感到\t很\t开心\t。 Chinese PROPN\tVERB\tADV\tADJ\tPUNCT (tgt) Tom\tfeels\tvery\thappy\t.
Output:	Vietnamese PROPN\tVERB\tADV\tADJ\tPUNCT

Table 10: An example of cross-lingual POS tagging task data, which use tab marks to split the words.

Instrcuct:	You are an expert in multilingual dependency parsing, learn the source language sentence's syntactic information and identify the target sentence's language type and parse it into the syntax format as follows. [Syntax format]: Each word has four columns separated by TAB, should follow the below rules: 1. Word index (starts from 1) 2. Original word form 3. Headword indices 4. Dependency type (*lowercase letters*)
Input:	(src) 汤姆\t感到\t很\t开心\t。 Chinese 1 \t汤姆 \t2 \tnsubj 2 \t感到 \t0 \troot 3 \t很 \t4 \tadvmod 4 \t开心 \t2 \txcomp 5 \t。 \t2 \tpunct (tgt) Tom\tfeels\tvery\thappy\t.
Output:	Vietnamese 1 \tTom \t2 \tnsubj 2 \tfeels \t0 \troot 3 \tvery \t4 \tadvmod 4 \thappy \t2 \txcomp 5 \t. \t2 \tpunct

Table 11: An example of cross-lingual dependency parsing task data, which use tab marks to split the words.

E Case Study on Using Label Banks

We design a three-stage method to optimize weak or moderate labels by prompting LLMs with dependency label banks. The following is an example of a label with weak memory strength:

Step 1: We first find the weak labels (cc) based on the memory strength and then obtain their corresponding POS tags.

ID	Word	POS	Head	Label	MS
1	I	PRON	2	nsubj	moderate
2	live	VERB	0	root	moderate
3	in	ADP	2	advmod	moderate
4	Hanoi	**PROPN**	2	**cc**	**weak**

Table 12: Dependency tree with weak memory label at index 4

Step 2: We use the current words' (Hanoi) POS tags (PROPN) to match their common head node words' POS tags (VERB), dependency labels (obl) and corresponding examples in the source and target dependency label banks as follows.

Field	Vietnamese (tgt)	Chinese (src)
Bank	Vietnamese (tgt)	Chinese (src)
Word POS	PROPN	PROPN
Head POS	**VERB**	**VERB**
Dep Label	**obl**	**obl**
Example	He works in Shanghai.	他工作在上海。
Structure	works ← **obl** - Shanghai	工作 ← **obl** - 上海

Table 13: Dependency label examples

Step 3: Make a prompt based on the above matched information.

[Prompt]: PROPN → VERB is often labeled as "obl", e.g., "He works in Shanghai. [works(VERB)← **obl** — Shanghai (PROPN)]", "他工作在上海 [工作(VERB) ← **obl** — 上海(PROPN)]"

Step 4: The prompt is fed into the LLM for optimizing the parsing result.

ID	Word	POS	Head	Label
1	I	PRON	2	nsubj
2	live	VERB	0	root
3	in	ADP	2	advmod
4	Hanoi	**PROPN**	2	**obl**

Table 14: Optimized dependency parsing result from LLM