Deep Stable Learning for Cross-Lingual Dependency Parsing

HAIJIANG LIU, Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China and Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan, China
CHEN QIU, Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China and Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan, China
QIYUAN LI, Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China and Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan, China
MAOFU LIU, Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China and Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan, China
LI WANG, Institute of Scientific and Technical Information of China, Beijing, China and Key Laboratory of Rich-media Knowledge Organization and Service of Digital Publishing Content, National Press and Publication Administration, Beijing, China

Haijiang Liu and Chen Qiu contributed equally to this research.

This work was supported by the National Key Research and Development Program of China under Grants 2022YFC3300801, the Research Project of China Publishing Promotion Association under Grants 2025ZBCH-JYYB17, the Knowledge Innovation Program of Wuhan-Shugun Project under Grants 2023010201020409, the Natural Science Foundation of Hubei Province (CN) under Grant No. 2025AFB078, and Research on the method of public opinion perception of science and technology from the perspective of extreme position under QN2024-12. We want to thank the Wuhan Supercomputing Center and Key Laboratory of Rich-media Knowledge Organization and Service of Digital Publishing Content for computational support and technical assistance.

Authors' Contact Information: Haijiang Liu, Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, Hubei, China and Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System,Wuhan University of Science and Technology, Wuhan, Hubei, China; e-mail: alecliu@ontoweb.wust.edu.cn; Chen Qiu (Corresponding author), Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, Hubei, China and Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan, Hubei, China; e-mail: chen@wust.edu.cn; Qiyuan Li, Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, Hubei, China and Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan, Hubei, China; e-mail: vickyuan@ontoweb.wust.edu.cn; Maofu Liu, Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, Hubei, China and Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan, Hubei, China; e-mail: liumaofu@wust.edu.cn; Li Wang, Institute of Scientiic and Technical Information of China, Beijing, China and Key Laboratory of Rich-media Knowledge Organization and Service of Digital Publishing Content, National Press and Publication Administration, Beijing, China; e-mail: wl@istic.ac.cn; Jinguang Gu, Computer Science and Technology, Wuhan University of Science and Technology,Wuhan, Hubei, China and Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan, Hubei, China; e-mail: simon@ontoweb.wust.edu.cn.

ACM 2375-4699/2025/06-ART56
https://doi.org/10.1145/3735509

JINGUANG GU, Computer Science and Technology, Wuhan University of Science and Technology, Wuhan, China and Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan, China

The Cross-lingual Dependency Parsing (XDP) task poses a significant challenge due to the differences in dependency structures between training and testing languages, known as the out-of-distribution (OOD) problem. Our research delved into this issue in the XDP dataset by selecting 43 languages from 22 language families. We found that the primary factor of the OOD problem is the unbalanced length distribution among languages. To address the impact of the OOD problem, we propose deep stable learning for Cross-lingual Dependency Parsing (SL-XDP), which utilizes deep stable learning with a feature fusion module. In detail, we implemented five feature fusion operations for generating comprehensive representations with dependency relations and the deep stable learning algorithm to decorrelate dependency structures with sequence length. Our experiments on Universal Dependencies have demonstrated that SL-XDP can lessen the impact of the OOD problem and improve the model generalization among 21 languages, with a maximum improvement of 18%.

CCS Concepts: • **Computing methodologies** → *Neural networks*; **Artificial intelligence**; **Natural language processing;**

Additional Key Words and Phrases: Cross-lingual dependency parsing, deep stable learning, out-of-distribution problem

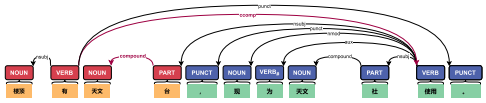
ACM Reference Format:
Haijiang Liu, Chen Qiu, Qiyuan Li, Maofu Liu, Li Wang, and Jinguang Gu. 2025. Deep Stable Learning for Cross-Lingual Dependency Parsing. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 24, 6, Article 56 (June 2025), 34 pages. https://doi.org/10.1145/3735509

1 Introduction

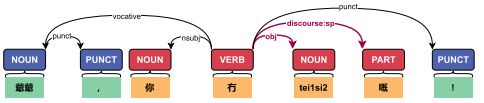
Cross-lingual Dependency Parsing (**XDP**) is a crucial process for Sentiment Analysis [15, 48] and Information Extraction [16, 37], requiring models to analyze semantic and grammatical structures in multiple languages [8]. As multilingual inputs are complex, the field has focused on developing a generalized dependency parser for many languages. Some studies have aimed to improve model transferability in **low-resource languages** (**LRL**) by modifying fine-tuning strategies on **High-Resource Languages** (**HRL**) for **multilingual Pre-trained Language Models** (**mPLMs**) [10]. Two common approaches are: (i) Fine-tuning with extra adaptation modules [41], where adapters are added to learn task-specific knowledge, and (ii) Multi-task fine-tuning (van der Goot et al. [42]; de Lhoneux et al. [9]), where training languages are treated as tasks and selected either statically or dynamically.

Although the solutions mentioned above achieve high model quality, results from de Lhoneux et al. [9] indicate that the model still fails to achieve generalization on LRL with an average accuracy of 36.4%, and the insufficiency of certain LRLs may be due to the model's inability to handle the **out-of-distribution** (**OOD**) problem. Figure 1 illustrates a sample of this problem in the XDP dataset Universal Dependencies [25] datasets. HRL datasets (like the Chinese dataset) often contain longer sentences (like Figure 1(a)) with more complex dependency structures. But sentences in LRL datasets are considerably shorter, and dependency structures are more straightforward (like Figure 1(b)). The model fine-tuned on HRL datasets performs better on complex dependency structures while failing to predict simple ones in LRL datasets.

In this article, we first investigate the OOD problem between HRLs and LRLs in the most commonly used multilingual dataset, Universal Dependencies, and the possible impact on the



(a) Dependency parsing tree from the Chinese dataset: The sentence "The rooftop has an observatory" has a complex structure due to the role of the object phrase in the following sub-sentence.



(b) Dependency parsing tree from the Cantonese dataset: The sentence "You (Grandpa) have 'tei1si2'" has a simpler structure.

Fig. 1. The demonstration of the OOD problem in the Universal Dependency datasets. Both examples highlight differences in dependency structures (bold red arrows) between the central verb and the object phrase.

XDP models (See Section 2). The conclusion is that the OOD problem between HRLs and LRLs originates from an imbalanced length distribution, causing unexpected dependency structures. The motivation of our method is to remove the correlation of length with relevant factors like grammar rules, further improving the model's performance.

Then, we expand the advantages of deep stable learning [49] to resolve the OOD problem in the XDP dataset because it can incorporate the heterogeneous influence brought by the OOD problem and enhance the model's generalization. As a solution, our deep **S**table **L**earning for **Cross**-Lingual **D**ependency **P**arsing (**SL-XDP**) has the main component called deep stable learning with Feature Fusion module includes a feature fusion step and a deep stable learning algorithm. Multiple feature fusion operations are presented in the feature fusion procedure to obtain a detailed feature representation for deep stable learning in NLP tasks. The deep stable learning algorithm calculates a learnable weight for each data sample and applies the reweighting method to identify and eliminate hidden data distribution correlations that reduce the impact of the OOD problem. Our experiments on Universal Dependencies demonstrate that the SL-XDP approach can help improve model generalization across more languages.

In this article, we mainly focus on three research objectives:

(1) To investigate the OOD problem between HRLs and LRLs in the Universal Dependencies dataset.
(2) To reduce the impact of the OOD problem in the XDP dataset by proposing the SL-XDP model and improving generalization among languages.
(3) To demonstrate the effectiveness of the SL-XDP model by conducting thorough experiments on the Universal Dependencies of 43 languages across 22 language families.

The main contributions of this article are summarized as follows:

(1) We investigate the OOD problem among languages in the XDP task and conclude the impact on the current XDP models.
(2) We propose SL-XDP, a novel model that lessens the impact of the OOD problem in the XDP task. The core procedure, deep stable learning with Feature Fusion module, comprises two main components: a feature fusion procedure and a deep stable learning algorithm. The feature fusion procedure computes comprehensive representations for training samples, and the deep stable learning algorithm dynamically updates the sample weight to reduce the impact of the OOD problem and then achieves better generalization among languages.
(3) Extensive experiments show that SL-XDP achieves improvements on the Universal Dependencies dataset. We further qualitatively analyze the quantity of gain of SL-XDP for different mPLMs and accurately list improvements in various languages.

2 OOD Problem in Universal Dependencies Dataset

We first give a thorough analysis of the OOD problem in the XDP dataset and then infer the impact on the current XDP models. Generally, the OOD problem in the UD dataset is that different languages have different length distributions, making it hard for models to model the dependency structures accurately. The details are as follows.

The primary focus of this work is to investigate the OOD problem in the Universal Dependency [25] dataset. For this study, we argue that the average lengths of datasets in various languages are the overall indicator of the complexity of the dependency structures. To verify it, we first find sentences expressing similar semantic structures from different languages. And compare the dependency structures, only including the words that contribute to this semantic structure. As illustrated in the previous section, we extract two sentences from the Chinese and Cantonese datasets (Figure 1). The two sentences express the semantics of "A has B" using the words in red blocks. The dependency structure of the Chinese sentences (Figure 1(a)) is more complex than the structure of the sentence in Cantonese (Figure 1(b)). It is straightforward to infer that the dependency structure of the Chinese sentence is more complicated than the one in the Cantonese sentence, since the object phrase in the Chinese sentence has roles in the sub-sentence separated by a comma. We sampled multiple sentences and found that longer sentences have more dependency relations and a more complicated dependency structure than shorter ones. Furthermore, the conclusion was consistent across all languages surveyed: longer sentences contain more complex dependency structures.

Once the connection between sentence lengths and dependency structures is established, we analyzed the average sentence lengths of both the HRL and LRL datasets. The results, shown in Figure 2, indicate that the HRL datasets typically contain longer sentences than the LRL datasets. Consequently, the sentences in the HRL datasets have more intricate dependency structures than those in the LRL datasets.

We argue that the imbalanced distribution of sentence lengths between HRL and LRL causes the difficulty of model transferring in XDP models. Previous studies (van der Goot et al. [42]; de Lhoneux et al. [9]) show that models tend to struggle to perform well in LRL datasets in zero-shot settings with an average accuracy of around 30%. The models fine-tuned to predict complex dependency structures in HRL datasets have limited capacity to model the dependency structures of short sentences in LRL datasets.

To this end, we introduce an SL-XDP model that incorporates deep stable learning with a feature fusion module. Through our proposed methodology, we strive to enhance the model's generalization by mitigating the OOD problem.

3 Related Work

In this section, we first briefly state the significant works in XDP and present the general development of deep stable learning theory.

3.1 Cross-Lingual Dependency Parsing

XDP is a technique that analyzes the grammatical and semantic structures of sentences in multiple languages. This can be useful for other tasks in natural language processing, such as Coreference Resolution and Information Extraction. To achieve accurate results, XDP models must predict the correct dependency relations between words in different language settings. Researchers have developed three main approaches to improve the generalization of dependency parsers across languages [8]: (1) Annotation projection [32, 44], which involves using parallel corpora to project annotations like POS taggings across languages. (2) Treebank translation (Tiedemann et al. [39];

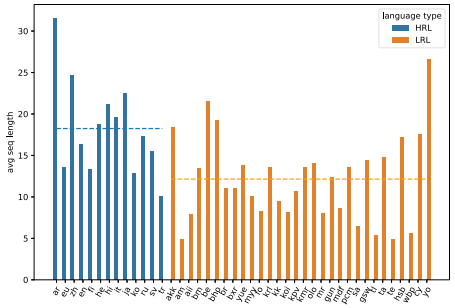


Fig. 2. The statistics report of the average sentence length for each language dataset in Universal Dependencies v2.5. The blue bars are datasets for HRL, and the orange bars are datasets for LRL. The lines in the middle are the average sentence lengths of HRL (blue) and LRL (orange), respectively. The languages are in code form for simplicity, and the mapping of language and language code is in the Appendix B.1.

Tiedemann and Agic [38]), which involves using machine translation systems to translate data to other languages. (3) Model transfer (Guo et al. [17], Eronen et al. [14]), in which models trained on HRLs are applied directly to predict dependency structures for sentences in LRLs. With the development of mPLMs, XDP models have achieved higher accuracy in various languages due to their transfer abilities [43].

Recent research has been focused on improving mPLMs by optimizing fine-tuning strategies. One approach is to incorporate additional parameters (adapters) into the original architecture to learn parsing knowledge from multiple languages during fine-tuning, like Üstün et al. [41] and Langedijk et al. [22]. This helps to reduce the impact of "catastrophic forgetting" [40] and enables the model to store task-specific knowledge in these adapters. One of the successes in this line of work is proposed by Üstün et al. [41], which enables the model to learn adapter parameters through language embeddings. Another approach is to view multilingual **dependency parsing** (**DP**) as multi-task learning with each language as an individual task (de Lhoneux et al. [9]; Wu and Dredze [43]; van der Goot et al. [42]). These methods improve the model transfer abilities with selected training samples, both static [43] and dynamic [9], which can reduce the computational expenses while expanding model generalization. Additionally, a powerful toolkit, MaChAmp [42], has been developed for multiple NLP tasks, which includes a smooth sampling strategy to prevent the model from being overwhelmed by large-sized datasets. A more recent approach, WAACL [9], adopts Worst-Case Aware Automated Curriculum Learning to dynamically select training samples and improve model accuracy on marginal languages. The integration of these two types of work has also been conducted by Pfeiffer et al. [27], where they presented an adapter-based framework that enables high-quality transfer to a wide range of tasks and languages.

Despite the existence of a significant amount of related work, the OOD problem continues to persist. To overcome this challenge, we infused our XDP model with a deep stable learning mechanism. This integration has significantly improved the model's generalization capabilities across multiple languages.

3.2 Deep Stable Learning

The deep stable learning theory deals with situations where the testing distribution could have unknown shifts from the training distribution due to complex data generation processes such as selection biases and confounding factors [4]. This can result in inaccurate predictions from models [35]. To solve this problem, the theory utilizes sample reweighting [35] to remove correlations between relevant and confounding factors in input data.

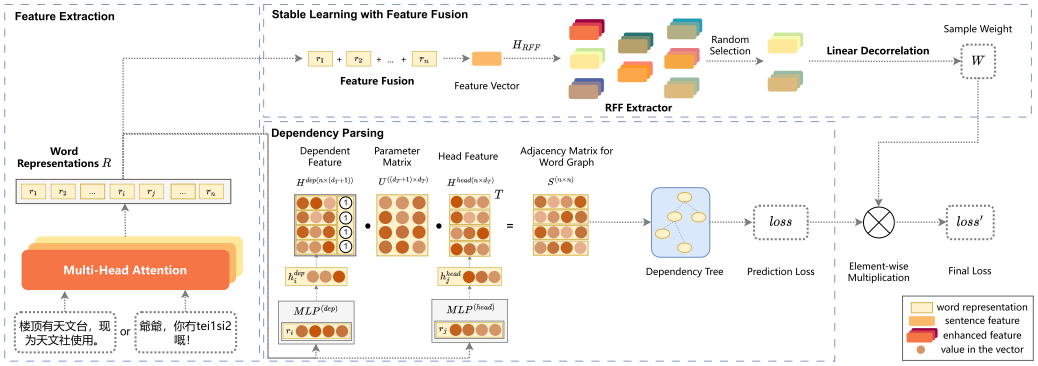


Fig. 3. The general structure of our deep stable learning model for XDP consists of three components: (1) The Feature Extraction procedure, which incorporates a contextual language model. (2) The DP module utilizes a graph-based parser. (3) Deep stable learning with a Feature Fusion module, which implements the feature fusion procedure to build a comprehensive representation and the deep stable learning algorithm to learn the sample weight.

There are several ways to use sample reweighting for linear models, such as CRLR [33] for computer vision, DGBR [20], DWR [21], and DVD [34] for linear regression problems. SRDO [35] is also an option for regression and classification problems. These methods aim to decrease linear correlations among variables. For non-linear models, StableNet [49] can be used for deep neural networks in image classification. It uses **Random Fourier Features** (**RFF**) and the cross-covariance matrix to achieve non-linear independence of input features.

Based on these successful approaches, we have applied the deep stable learning method to XDP models. This helps to reduce the impact of OOD problems between HRL and LRL.

4 Methodology

Here we outline the XDP problem. The goal is to produce a tree structure, denoted as $y' = (W, E; S^{(tree)})$, from a sentence $S = [w_0, w_1, \ldots, w_n]$, where n represents the length of the sentence. The tree structure is made up of words, their dependencies, and a score, represented by W, E, and $S^{(tree)}$, respectively. This problem can be formulated as follows: $y' = \arg\max_s F(S, \Theta)$. In this equation, F refers to our SL-XDP, and Θ represents the model parameters.

As shown in Figure 3, our proposed SL-XDP framework consists of three components. (1) The Feature Extraction module converts input sentences in different languages into word representations. (2) The DP module uses a biaffine attention neural network and parsing algorithms to process the word representations and obtain the prediction loss. (3) The deep stable learning with the Feature Fusion module calculates the sample weight using a feature fusion procedure with the word representations, multiplies the learnable weight with the prediction loss, and updates the model parameters with the final loss.

Algorithm 1 demonstrates the entire training process of one input sentence during one epoch. Each module will be presented in the corresponding section.

4.1 Feature Extraction and Dependency Parsing

In our study, the Feature Extraction module is responsible for converting input sentences into word representations, which are essential for downstream tasks like DP. The contextual language model called mBERT, as described by Devlin et al. [10], extracts word representations from input

ALGORITHM 1: SL-XDP for one sentence input in one epoch

Input: Input sentence $S = [w_1, w_2, \ldots, w_n]$
Output: Final loss for update model parameters loss$' \in \mathbb{R}$
Initialization:

(1) Encoder parameters $\Theta_{PLM} \leftarrow$ Pre-train Language Model
(2) Biaffine Neural Network parameters $\Theta_{DP} \leftarrow$ random initial
(3) Feature Fusion module parameters $\Theta_{FF} \leftarrow$ random initial
(4) deep stable learning weight $\mathbf{w} \leftarrow 1$

Feature Extraction: Extract token vectors from the input sentence S using the Pre-trained Language Model:
$\mathbf{R} \leftarrow$ `FeatureExtraction`$(S; \Theta_{PLM})$ (Refer to Equation (1))
Dependency Parsing: Parse the token vectors and compute the prediction loss using the Biaffine Neural Network:
loss $\leftarrow$ `DependencyParsing`$(\mathbf{R}; \Theta_{DP})$ (Refer to Equations (2)–(4))
Feature Fusion: Fuse the token vectors to generate the sentence representation:
$\mathbf{f} \leftarrow$ `FeatureFusion`$(\mathbf{R}; \Theta_{FF})$
Five fusion methods are provided:

- Simple Add (Equation (5))
- Weighted Add (Equation (7))
- Multi-Head Self-Attention (Equation (9)–(11))
- Attention-Free (Equation (12))
- LAFF (Equation (13))

deep stable learning: Perform deep stable learning to iteratively calculate the sample weight $\mathbf{w}$:
$\mathbf{w} \leftarrow$`StableLearning`$(\mathbf{f}; \mathbf{w})$
Use the following techuquies:

- RFF Augmentation (Equation (16))
- Decorrelation Weight Learning (Equation (20))
- Saving and Reloading (Equations (22) and (23))

Calculate Final Loss: Combine the computed loss and deep stable learning weight:
loss$' \leftarrow \mathbf{w} *$ loss
return loss$'$

sentences. We employed WordPiece tokenization to tokenize the sentence, which was then passed through the embedding layers to obtain the representation. This representation is denoted as $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n]$, where $\mathbf{r}_i \in \mathbb{R}^{d_T}$ represents the ith word's representation and d_T is the word representation dimension. The token representations are modeled as follows:

$$\mathbf{r}_i = \mathrm{LM}(w_i; \Theta), \tag{1}$$

where w_i denotes the ith word of the input sentence, H is the tokenization dimension and the language model parameter is Θ, and language model is LM for simplicity.

The DP module uses these word representations to predict the syntactic structure of the sentence, which is crucial for understanding the relationships between words. We take advantage of the graph-based biaffine attention classifier [12] to improve the modeling of the dependency relation, which is an asymmetric binary relation indicating that tokens can serve as the dependent or the head of the relation. The classifier comprises two feedforward layers (MLP) and a biaffine attention procedure, which use two MLPs to compute the representation of a word being the dependent $\mathbf{h}_i^{(\mathrm{dep})} \in \mathbb{R}_{d_T}$ or the head $\mathbf{h}_i^{(\mathrm{head})} \in \mathbb{R}_{d_T}$ of a dependency relation from the word representation $\mathbf{r}_i$ respectively as follows:

$$\begin{aligned} \mathbf{h}_i^{(\mathrm{dep})} &= \mathrm{MLP}^{(\mathrm{dep})}(\mathbf{r}_i) \\ \mathbf{h}_i^{(\mathrm{head})} &= \mathrm{MLP}^{(\mathrm{head})}(\mathbf{r}_i) \end{aligned} \tag{2}$$

We summarize the dependent and head features into two matrices $\mathbf{H}^{(\text{dep})} \in \mathbb{R}_n^{(d_T+1)}$ and $\mathbf{H}^{(\text{head})} \in \mathbb{R}_n^{d_T}$ separately and calculate the score of the relation between every two words in the sentence through the biaffine attention procedure presented using the following procedure:

$$\begin{aligned} \mathbf{H} &= \text{Concate}_{i=0}^{n}(\mathbf{h}_i) \\ \mathbf{S}^{\text{arc}} &= \mathbf{H}^{(\text{dep})} \cdot \mathbf{U} \cdot \mathbf{H}^{(\text{head})^{\mathsf{T}}} \end{aligned} \tag{3}$$

where Concate means stack all vectors together to create a matrix.

During fine-tuning, we update the parameter matrix $\mathbf{U} \in \mathbb{R}_{(d_T+1)}^{d_T}$ to obtain a more accurate score matrix $\mathbf{S}^{\text{arc}} \in \mathbb{R}_n^n$, which is used to find the highest-scoring dependency tree y' from the score matrix as our output using the Chu–Liu/Edmonds algorithm (Chu and Liu [6]; Edmonds et al. [13]).

Finally, the model calculates the prediction loss between ground truths y and the model prediction y' by the cross-entropy function L using:

$$\text{loss} = L(y, y') \tag{4}$$

4.2 Deep Stable Learning with Feature Fusion

The deep stable learning with the Feature Fusion module ensures that the model is robust to OOD data by calculating sample weights that adjust the loss function, thereby improving the model's generalization ability. First, the representations of words $\mathbf{R}$ are sent in the feature fusion module to create a complete representation for the input sentence. Then, the sentence representation is passed to the deep stable learning module to calculate the sample weight for the input sentence. Finally, the sample weight multiplies the corresponding loss to obtain the final loss, which updates the model parameters.

4.2.1 Feature Fusion Module. Our learning algorithm needs a sample representation that covers enough information for the task. In image classification [49], the sample representation was the feature map of one picture. However, our XDP model converts input samples into a series of token vectors using pre-trained language models. The special token [CLS] in Devlin et al. [10] represents one sentence, but it was created for sentence-level classification tasks and has no relevance to the DP task. Since XDP requires a smaller granularity of information, such as the relationship between words, we use feature fusion operations to include word relation information instead of using [CLS] representations.

Feature fusion functions allow models to merge information in different feature vectors and build a more comprehensive representation, which has been widely implemented in the multimodal research field. The primary operations are the add and concatenate functions without information loss. Other implementations using deep neural networks can filter irrelevant information, leading to more focused representations [18]. In this article, we investigate the effectiveness of five feature fusion operations, demonstrated in Figure 4, and take the simple add as our final operation.

Simple add. We demonstrate the "**simple add**" operation, where adding all token vectors together to form the representation of one sentence:

$$\mathbf{f} = \sum_{i=1}^{n} \mathbf{r}_i \tag{5}$$

where $\mathbf{r}_i$ denotes the representation of the ith words in the input sentence, and $\mathbf{f} \in \mathbb{R}^{d_T}$ denotes the sentence representation.

Example from training data: Consider a sentence with three words: "The cat sat." Suppose the word vectors for "The", "cat ,and "sat" are $\mathbf{r}_1 = [1, 2, 3]$, $\mathbf{r}_2 = [4, 5, 6]$, and $\mathbf{r}_3 = [1 + 4 + 7, 2 +$

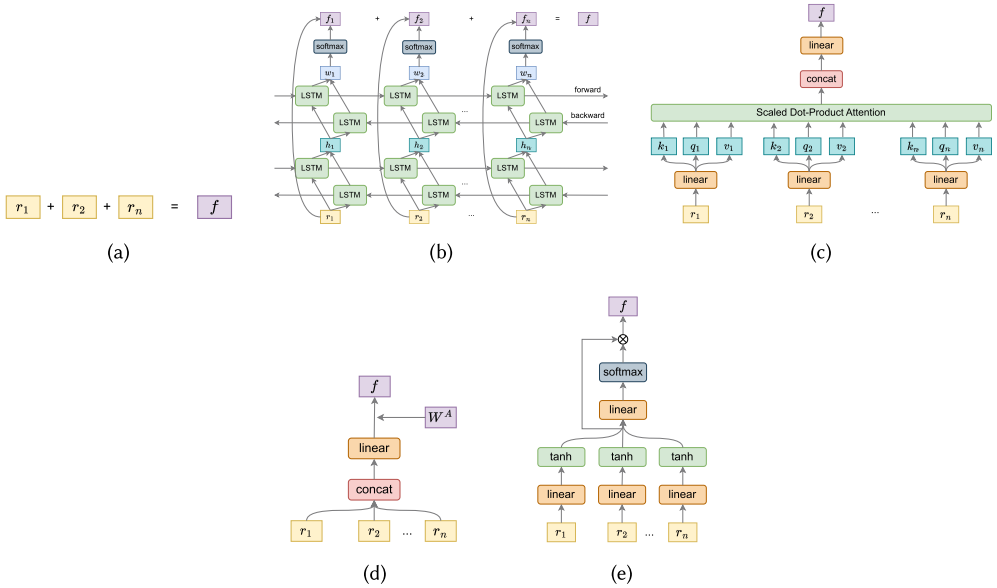


Fig. 4. We implement five feature fusion operations, namely (a) Simple add, (b) Weighted add, (c) Multi-Head Self-Attention, (d) Attention-free, and (e) LAFF [18].

$5 + 8, 3 + 6 + 9] = [12, 15, 18] = [7, 8, 9]$, respectively. The sentence representation $\mathbf{f}$ would be

$$\mathbf{f} = \mathbf{r}_1 + \mathbf{r}_2 + \mathbf{r}_3 = [1 + 4 + 7, 2 + 5 + 8, 3 + 6 + 9] = [12, 15, 18] \tag{6}$$

We use the primary operation "Simple add" for two considerations: (1) Adding the word vectors together for a single sentence does not create new training objectives, enabling our XDP model to concentrate on the two existing targets of learning model parameters in mBERT and the Biaffine parameter matrix. (2) Secondly, the "add" operation does not increase the already enormous memory requirement of our XDP model framework, MaChAmp [42].

Weighted add. The input tokens in the dependency task have varying significance, so we utilized a **BiLSTM** neural network to create a learnable weight. The hidden states from a BiLSTM network are applied with a softmax function to ensure the weight value ranges from 0 to 1:

$$\begin{aligned} W_f &= \sigma(\text{BiLSTM}(\mathbf{R})) \\ \mathbf{f} &= \sum_{i=1}^{n} w_i \mathbf{r}_i \end{aligned} \tag{7}$$

where $w_i \in W_f$ and σ is the softmax activation function.

Example from training data: Using the same sentence "The cat sat", suppose the BiLSTM generates weights $w_1 = 0.2, w_2 = 0.5$, and $w_3 = 0.3$ or the words "The", "cat", and "sat" , respectively. The weighted sentence representation $\mathbf{f}$ would be

$$\mathbf{f} = w_1\mathbf{r}_1 + w_2\mathbf{r}_2 + w_3\mathbf{r}_3 = 0.2 \cdot [1, 2, 3] + 0.5 \cdot [4, 5, 6] + 0.3 \cdot [7, 8, 9] = [4.3, 5.3, 6.3] \tag{8}$$

Our implementation only includes two layers of BiLSTM to keep the number of learnable parameters small while still maintaining effective modeling.

Multi-Head Self-Attention (MHSA). As the cornerstone of Transformers, MHSA combines information from all token features by multiplying the blending weights using a self-attention

mechanism called QKV. Inspired by the success of Transformers-based models, we take 1 layer of self-attention to extract sentence representation from every token.

$$\begin{aligned} Q_i &= \text{linear}(r_i) \\ K_i &= \text{linear}(r_i) \\ V_i &= \text{linear}(r_i) \end{aligned} \tag{9}$$

Next, the scaled dot-product attention is built at the top to merge information from all attention heads:

$$\begin{aligned} &\text{Attention}(Q, K, V) = \sigma\left(\frac{QK^T}{\sqrt{d_k}}\right)V \\ &\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, \ldots, head_h)W^O \end{aligned} \tag{10}$$

where $\sqrt{d_k}$ is the dimension of the key vector k and query vector q, $head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

Finally, we concatenate all the heads and send them to a linear layer to generate the sentence representation:

$$\mathbf{f} = \text{linear}(\text{MultiHead}(Q, K, V)) \tag{11}$$

Example from training data: Suppose we have two attention heads. For the word "cat", the first head might focus on the subject-verb relationship, while the second head focuses on the object-verb relationship. The final sentence representation $\mathbf{f}$ would be a weighted combination of these attention heads.

The MHSA method transforms token vectors to be more contextually relevant and has stronger long-term memory compared to BiLSTM. However, it requires more computational cost.

Attention-Free. The "Attention-Free" fusion operation does not require calculating the attention matrix, resulting in lower computational costs while preserving long-term memory capacity. This is accomplished by substituting the self-attention layer with a trainable attention-free weight matrix. The weighted features are then passed through a linear layer to create the sentence representation:

$$\mathbf{f} = \text{linear}(\mathbf{R}\mathbf{W}^A), \tag{12}$$

where $\mathbf{W}^A$ is the randomly initialized attention-free weight matrix.

Example from training data: If W^A is a matrix that assigns equal importance to all tokens, the sentence representation $\mathbf{f}$ would be a simple linear transformation of the token vectors.

The operation that doesn't require much attention is more cost-effective and simpler to use than MHSA, yet it still effectively extracts information from each token.

LAFF. Following Hu et al. [18], we use their lightweight fusion block called LAFF. It uses a linear layer to encode the token feature into a weight:

$$\begin{aligned} \mathbf{r}'_i &= \sigma(\text{linear}(\mathbf{r}_i)) \\ a_1, \ldots, a_n &= \sigma(\text{linear}(\mathbf{r}'_1, \ldots, \mathbf{r}'_n)) \end{aligned} \tag{13}$$

where $a_1, \ldots, a_n$ represents the encoded weight sequence.

Then it multiplies the weight matrix by the token vectors as the final sentence representation.

$$\mathbf{f} = \sum_{i=1}^{n} a_i \mathbf{r}_i \tag{14}$$

Example from training data: If the encoded weights for "The", "cat", and "sat" are $a_1 = 0.1$, $a_2 = 0.6$, and $a_3 = 0.3$ respectively, the sentence representation $\mathbf{f}$ would be

$$\mathbf{f} = a_1\mathbf{r}_1 + a_2\mathbf{r}_2 + a_3\mathbf{r}_3 = 0.1 \cdot [1, 2, 3] + 0.6 \cdot [4, 5, 6] + 0.1 \cdot [7, 8, 9] = [4.6, 5.6, 6.6] \tag{15}$$

LAFF has fewer trainable parameters and is more efficient to compute.

4.2.2 Deep Stable Learning. The deep stable learning mechanism we introduce utilizes RFF to augment sample representations into higher dimensions. It also uses a technique to eliminate the correlation between random variables in the enhanced feature vectors by setting the Frobenius norm of the weighted partial cross-covariance matrix to zero. Furthermore, we incorporate saving-reloading strategies to simulate the global weight during training using **Stochastic Gradient Descent** (**SGD**).

RFF augmentation. Deep stable learning aims to eliminate correlations between relevant and confounding factors, which are random variables in the feature vectors. The Frobenius norm of the weighted partial cross-covariance matrix [21] can be zeroed out to remove linear correlations between two variables. However, non-linear relations are considerably complicated, and the linearity of these correlations in XDP datasets may not be obvious. Therefore, we use the RFF [31] to expand the characteristics of all variables into higher dimensions as follows, which allows for the decorrelation of both linear and non-linear relations:

$$\mathcal{H}_{\mathcal{RFF}} = \{h : \mathbf{f}'_i \rightarrow \sqrt{2}\cos(\omega\mathbf{f}_i + \phi) | \omega \sim N(0, 1),\ \phi \sim \text{Uniform}(0, 2\pi)\}, \tag{16}$$

where $\mathbf{f}_i \in \mathbb{R}^{d_T}$ denotes the ith sentence representation, $\mathbf{f}'_i \in \mathbb{R}^{d_T}$ denotes the enhanced representation of the ith input sentence, ω is sampled from Normal distribution $N(0, 1)$, and ϕ is sampled from the Uniform distribution $U(0, 2\pi)$.

Decorrelating weight learning. The core procedure of deep stable learning is to eliminate the relations between any pair of variables $\mathbf{Z}_{:,i} \in \mathbb{R}^{d_T}$ and $\mathbf{Z}_{:,j} \in \mathbb{R}^{d_T}$ in the feature vectors. After RFF augmentation, we can eliminate the correlations in a linear fashion [49] under the following theorem. The Frobenius norm of the weighted partial cross-covariance matrix is limited to zero so that the hidden relations between variables can be decoupled.

Theorem 4.1. *If the product of two variable kernels k_A, k_B, where A, B representing the variables, is characteristic and $\mathbb{E}[k_A(A, A) < \infty]$, $\mathbb{E}[k_B(B, B) < \infty]$.*

$$A \text{ and } B \text{ are independent} \iff \Sigma_{AB} = 0 \tag{17}$$

where Σ_{AB} is the cross-covariance operator.

First, we use $\mathbf{W} \in \mathbb{R}^b_+$ to denote the sample weight and $\sum_{i=1}^b \mathbf{w}_i = b$, where b is the number of samples and the calculation of the partial cross-variance matrix is

$$\hat{\Sigma}_{AB;\mathbf{W}} = \frac{1}{b-1}\sum_{p=1}^{b}\left[\left(\mathbf{w}_p\mathbf{u}(A_p) - \frac{1}{b}\sum_{k=1}^{b}\mathbf{w}_k\mathbf{u}(A_k)\right)^T * \left(\mathbf{w}_p\mathbf{v}(B_p) - \frac{1}{b}\sum_{k=1}^{b}\mathbf{w}_k\mathbf{v}(B_k)\right)\right], \tag{18}$$

where A and B are representing the factors $\mathbf{Z}_{:,i}$ and $\mathbf{Z}_{:,j}$, $A_i \in \mathbb{R}^{d_T}$ and $B_i \in \mathbb{R}^{d_T}$ are sampled from the distribution of A and B.

The definition of $\mathbf{u}$ and $\mathbf{v}$ are as follows:

$$\begin{aligned}\mathbf{u}(A) &= \left(u_1(A), u_2(A), \ldots, u_{n_A}(A)\right), u_j(A) \in \mathcal{H}_{\mathcal{RFF}}, \forall j \\ \mathbf{v}(B) &= \left(v_1(B), u_2(B), \ldots, v_{v_B}(B)\right), v_j(B) \in \mathcal{H}_{\mathcal{RFF}}, \forall j\end{aligned} \tag{19}$$

where we select n_A, n_B functions from $\mathcal{H}_{\mathcal{RFF}}$ respectively in Equation (7).

Secondly, our optimization objective is to learn the weight zeroing out the Frobenius norm of the partial cross-covariance matrix ($||\hat{\Sigma}_{\mathbf{AB};\mathbf{W}}||_F^2$) by

$$\mathbf{W}^* = \text{argmin}_{\mathbf{W}\in\Delta_b} \sum_{1<i<j<d_T} \left\|\hat{\Sigma}_{\mathbf{AB};\mathbf{W}}\right\|_F^2 \tag{20}$$

where $\Delta_b = \{\mathbf{W} \in R_+^b | \sum_{i=1}^b \mathbf{w}_i = b\}$.

Eventually, the algorithm iteratively optimizes the sample weight $\mathbf{W}$ and updates the model parameters Θ through gradient descent by

$$\begin{aligned} F^{(t+1)} &= \arg\min_F \sum_{i=1}^{n} \mathbf{w}_i^{(t)} L\left(F\left(S_i\right), y_i; \Theta\right), \\ \mathbf{W}^{(t+1)} &= \arg\min_{\mathbf{W}\in\Delta_b} \sum_{1<i<j<m_{\mathbf{Z}}} \left\|\hat{\Sigma}_{\mathbf{Z}_{:,i}^{(t+1)}\mathbf{Z}_{:,j}^{(t+1)};\mathbf{W}}\right\|_F^2 \end{aligned} \tag{21}$$

where $\mathbf{Z}^{(t+1)} = F^{(t+1)}(S)$ and $L(.,.)$ representing the cross-entropy loss function, S_i and y_i are the ith input sentence and ground truth, and t denotes the tth step. The sample weight is initialized as an all-ones matrix.

Saving and reloading. Our deep stable learning algorithm is designed to calculate sample weights on a global scale, removing relationships between variables in the entire dataset. The learnable sample weight, represented by $\mathbf{W}^* \in \mathbb{R}_+^b$, considers all sentences in the training data, meaning that $b = n$. However, our model uses SGD optimization, which only observes a portion of the sentences in each batch. This makes it difficult to learn the global weight for all training sentences. To address this issue, we employ the saving and reloading technique to simulate the weight globally. This involves saving the sentence representations and sample weights during training and then reloading them as global information to optimize sample weights.

Specifically, after completing one training batch, the model stores the sample weight and sentence representations, then concatenates the saved vectors as the new information for the coming batch by

$$\begin{aligned} \mathbf{Z}_O &= \text{Concat}\left(\mathbf{Z}_{G_1}, \mathbf{Z}_{G_2}, \ldots, \mathbf{Z}_{G_k}, \mathbf{Z}_L\right), \\ \mathbf{W}_O &= \text{Concat}\left(\mathbf{W}_{G_1}, \mathbf{W}_{G_2}, \ldots, \mathbf{W}_{G_k}, \mathbf{W}_L\right) \end{aligned} \tag{22}$$

where $\mathbf{Z}_{G_i} \in \mathbb{R}_b^{d_T}$ and $\mathbf{W}_{G_i} \in \mathbb{R}^b$ denote the global feature and weight in the ith batch and b is the batch size. $\mathbf{Z}_L \in \mathbb{R}_b^{d_T}$ and $\mathbf{W}_L \in \mathbb{R}^b$ denote the local information for the current batch, $\mathbf{Z}_O$ and $\mathbf{W}_O$ are the new information sent to the new batch.

Before finishing the training on all data, we keep $\mathbf{W}_{G_i}$ frozen and learn the local weight $\mathbf{W}_L$ using Equation (14). At the end of a training epoch, the global and local information are merged and $(\mathbf{Z}'_{G_i}, \mathbf{W}'_{G_i})$ takes the place of every $(\mathbf{Z}_{G_i}, \mathbf{W}_{G_i})$ into the new epoch:

$$\begin{aligned} \mathbf{Z}'_{G_i} &= \alpha_i \mathbf{Z}_{G_i} + (1-\alpha_i)\mathbf{Z}_L, \\ \mathbf{W}'_{G_i} &= \alpha_i \mathbf{W}_{G_i} + (1-\alpha_i)\mathbf{W}_L \end{aligned} \tag{23}$$

where α is a sequence of smoothing parameters to handle both the long-term memory (α_i is large) and the short-term memory (α_i is small).

Table 1. Training Dataset Information Summary

Language	Treebanks	Train size	Valid size	Data size
Arabic	PADT	6075	909	680
Basque	BDT	5396	1798	1799
Chinese	GSD	3997	500	500
English	EWT	12543	2002	2077
Finnish	TDT	12217	1364	1555
Hebrew	HTB	5241	484	491
Hindi	HDTB	13304	1659	1684
Italian	ISDT	13121	564	482
Japanese	GSD	7125	511	550
Korean	GSD	4400	950	989
Russian	SynTagRus	48814	6584	6491
Swedish	Talbanken	4303	504	1219
Turkish	IMST	3664	988	983

The "treebanks" indicate the source of the sentences. The data size measures the number of sentences in the dataset.

5 Experiments and Results

Our SL-XDP is assessed using the widely used XDP dataset Universal Dependencies v2.3 and v2.5, using the Labeled Accuracy Score. We also use the following experimental setup to compare our results to three current XDP parsers.

Dataset. Our method is trained and evaluated on the multilingual DP dataset **Universal Dependencies** [25]. We chose the 2.3 version (UD v2.3) and the 2.5 version (UD v2.5) as our datasets. Both versions consist of nearly 200 treebanks in over 100 languages. Following the training paradigm designed by Üstün et al. [41], 13 topologically diverse HRL and 30 LRL are sampled as training languages and zero-shot testing languages, respectively. The used treebanks and the data size are listed in Tables 1 and 2. The length distribution of HRL and LRL is in Figure 2. For additional language information, such as language families and codes, please refer to Appendix B.1.

Evaluation metric. We evaluate performances using the **Labeled Accuracy Score** (**LAS**) on HRL and LRL data. It calculates the percentage of correct dependency relations (including labels) of all relations using:

$$\text{LAS} = \frac{N_{\text{correct}}}{N_{\text{all}}} \tag{24}$$

where N_{correct} is the number of relations that the model predicted correctly, and N_{all} is the total number of relations predicted by the model.

Baselines. The three recent XDP parsers are employed as baseline models using mBERT as their backbone.

UDapter [41], which is a powerful parser that uses extra adaptation modules and language embeddings to learn adapters. To conduct a comparable experiment, our model is trained on UD v2.3 since the software dependencies of UDapter are not available in the pip library, and the results are compared to those reported in their paper.

MaChAmp [42], which is an effective multi-tasking model based on the NLP library AllenNLP. It provides a smooth sampling strategy to prevent the model from being overwhelmed by large datasets.

Table 2. Zero-Shot Testing Dataset Information Summary

Language	Treebanks	Data size
Akkadian	PISANDUB	101
Amharic	ATT	1074
Assyrian	AS	57
Bambara	CRB	1026
Belarusian	HSE	253
Bhojpuri	BHTB	254
Breton	KEB	888
Buryat	BDT	908
Cantonese	HK	1004
Erzya	JR	1550
Faroese	OFT	1208
Karelian	KKPP	228
Kazakh	KTB	1047
Komi Permyak	UH	49
Komi Zyrian	LATTICE, IKDP	327
Kurmanji	MG	734
Livvi	KKPP	106
Marathi	UFAL	47
Mbya Guarani	THOMAS, DOOLEY	1144
Moksha	JR	65
Naija	NSC	948
Sanskrit	UFAL	230
Swiss German	UZH	100
Tagalog	TRG	55
Tamil	TTB	120
Telugu	MTG	146
Upper Sorbian	UFAL	623
Warlpiri	UFAL	55
Welsh	CCG	956
Yoruba	YTB	100

The "treebanks" indicate the source of the sentences. The data size measures the number of sentences in the dataset.

WAACL [9], which is a Worst-Case Aware Automated Curriculum Learning model designed to improve performance on marginal languages. It sends training languages according to the prediction loss during training to promote the model's performance on marginal languages.

To make a direct comparison, the last two models are re-trained on the HRL of UD v2.5, which includes more languages and removes multiple annotation errors. We then tested them on LRL datasets.

Experiment configuration. SL-XDP implements a transformer encoder mBERT [10] with an embedding dimension d_T of 768. The dropout is 0.3, the batch size b is 32, the training epoch is 80, and the initial learning rate is 1e-0.5. We initialize its weight using the *bert-base-multilingual-cased* checkpoint. For the dependency parser, the dependency arc and label vectors are in dimensions

Table 3. Labeled Attachment Scores (LAS,%) on the Subset of LRL for Models Fine-Tuning with Extra Adaptation Modules: UDapter [41], Models Fine-Tuning with Selected Samples: MaChAmp [42], WAACL [9], and SL-XDP

Type	Model	aii	am	be	bxr	cy	fo	gsw	hsb	mr	olo	pcm	ta	te	wbp	yo	yue	AVG
mBERT	UDapter†	14.3	5.9	79.3	28.9	54.4	69.2	45.5	54.2	44.4	43.3	36.7	46.1	71.1	12.1	42.7	32.8	36.5
	MaChAmp	5.3	4.1	81.5	24.6	55.0	66.5	43.3	50.1	56.7	41.1	33.5	52.1	70.3	6.5	39.6	31.7	34.9
	WAACL	7.9	20.2	78.1	25.8	55.4	67.6	43.9	50.6	60.4	40.9	33.7	52.4	69.9	6.2	38.3	32.8	36.4
	SL-XDP	8.4	22.1	78.5	26.0	55.5	67.0	48.4	50.7	60.1	41.1	33.4	52.3	70.6	6.9	37.8	33.1	36.5
	- w/o WAACL	4.0	4.8	81.4	24.8	54.9	66.6	43.5	50.3	55.1	40.8	34.3	52.3	70.2	5.9	39.7	31.9	34.8
XLM-R large	MaChAmp	4.0	67.8	86.5	25.8	59.8	72.3	43.4	58.7	67.8	53.0	38.5	52.4	76.1	11.8	2.6	43.1	41.6
	WAACL	2.9	68.4	85.3	27.3	59.9	72.5	44.9	61.4	68.6	56.5	38.8	54.0	75.9	7.8	3.8	41.8	41.8
	SL-XDP	2.9	68.4	85.3	27.5	59.9	72.6	45.2	67.5	68.9	56.1	38.7	53.9	76.1	7.5	3.5	41.8	41.7
	- w/o WAACL	2.9	67.0	86.2	25.8	59.7	72.7	40.9	60.2	68.6	53.0	37.5	52.7	74.6	12.1	2.4	42.4	41.1

The complete results of all 30 LRL are in Appendix C. Our ablation studies are conducted in two dimensions: mPLM backbones (mBERT and XLM-R large), and w/o WAACL. "AVG" is the average LAS of 30 LRL datasets. UDapter† model is trained on UD v2.3 in their paper.

768 and 256. In SL-XDP, the feature fusion weight is 1, the memory preserve rate α is 0.9, and the enhancement dimension of RFF is 10.

5.1 Results

The overall performance of our SL-XDP model with other recent baselines on LRL and HRL is shown in Tables 3 and 4, respectively. We have compared our model with three baselines using mBERT as backbone: UDapter† [41], which uses extra adaptation modules to learn task-specific knowledge, MaChAmp [42], which is a multi-tasking model built on the AllenNLP library, and WAACL, which is the Worst-case Aware Automated Curriculum Learning model proposed by [9]. Our model outperforms these baselines in the average LAS on LRL datasets by 1.4%, 0.1%, and 1.6%.

LRL performance. Table 4 displays the results for SL-XDP on the subset of LRL datasets. For the complete results, refer to Table C. Compared to WAACL, SL-XDP has improved 13 out of 30 languages, with an average LAS increase of 0.1%. Additionally, several languages have shown significant progress, such as MDF with a gain of over 1.4% LAS, and AM with a gain of over 2.1% LAS. In comparison to MaChAmp, SL-XDP performs better than 25 out of 30 languages. Particularly, AM shows a significant improvement of 18%. Compared to UDapter† on UD v2.3,[1] our model improves eight languages from UDapter† and has an average LAS gain of 1.4%. The most notable improvement is in AM, which has increased by 18.3%.

A few drops are observed compared to three baselines: The amplitudes of decrease compared to WAACL are all below 1.0%, and the largest descent is in GSW, with a reduction of 0.9%. As for comparing to MaChAmp, YO poses a challenge for the SL-XDP as its prediction accuracy decreases by 1.8%.

HRL performance. The results of SL-XDP on HRL datasets are presented in Table 4. SL-XDP has outperformed WAACL [9] in 8 out of 13 languages, resulting in an average LAS increase of 0.2%. Notably, significant improvements of 1.1% have been observed in languages that initially had poor scores, such as AR. Our SL-XDP model shows only a slight improvement of 1.6% in ZH and 4.5% in EN compared to the UDapter† model on UD v2.3.[2]

[1]Part of languages are not included in the latest UD v2.3 archive, so we have marked their results as "–" in Appendix C.
[2]We cannot fully analyze the results of UDapter† as their data may not have been cleaned by the tools provided by MaChAmp.

Table 4. Labeled Attachment Scores (LAS,%) on HRL for Models Fine-Tuning with Extra Adaptation Modules: UDapter [41], Models Fine-Tuning with Selected Samples: MaChAmp [42], WAACL [9], and SL-XDP

Type	Model	ar	eu	zh	en	fi	he	hi	it	ja	ko	ru	sv	tr	AVG
mBERT	UDapter†	84.4	89.7	83.2	83.3	89.0	88.8	92.0	93.5	92.8	85.9	92.2	90.3	69.9	87.3
	MaChAmp	82.8	82.7	85.7	89.8	88.9	84.2	91.8	93.2	92.9	85.6	92.4	89.9	68.7	86.8
	WAACL	67.3	80.0	74.3	83.2	72.8	54.4	85.6	85.8	90.3	76.6	86.0	82.7	56.0	76.5
	SL-XDP	68.4	80.2	74.3	83.3	72.7	54.5	85.8	85.8	90.4	76.7	86.1	83.0	56.0	76.7
	- w/o WAACL	83.1	82.3	84.0	89.8	88.8	85.4	91.8	92.6	94.2	83.5	91.7	87.4	66.3	86.2
XLM-R large	MaChAmp	85.8	87.5	87.2	92.0	94.4	88.5	93.2	94.5	94.0	89.4	94.8	93.8	73.8	89.9
	WAACL	84.4	86.8	86.3	91.4	93.3	88.2	93.0	93.4	92.8	88.9	93.6	93.2	72.8	89.1
	SL-XDP	84.3	86.8	86.3	91.4	93.4	88.2	93.0	93.5	92.8	88.9	93.7	93.3	72.9	89.1
	- w/o WAACL	85.8	87.6	87.5	91.9	94.3	88.7	93.2	94.5	94.0	89.7	94.8	93.6	73.7	90.0

Our ablation studies are conducted in two dimensions: mPLM backbones (mBERT and XLM-R large), and w/o WAACL. "AVG" shows the average LAS of HRL datasets. UDapter† model is trained on UD v2.3 in their paper.

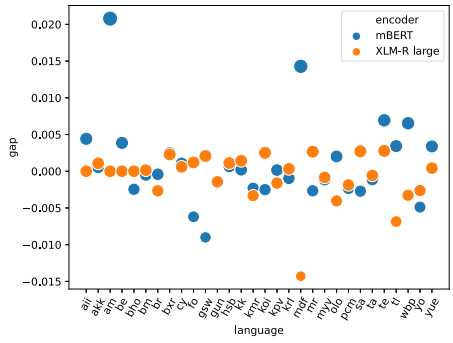
Although our model performs well, it does not match the performance of MaChAmp [42] on HRL. One possible reason is that the MaChAmp model has overfitted the HRL datasets, and unfortunately, SL-XDP was not able to assist the model any further. To investigate this, the ablation study is discussed in Section 5.2.

5.2 Analysis

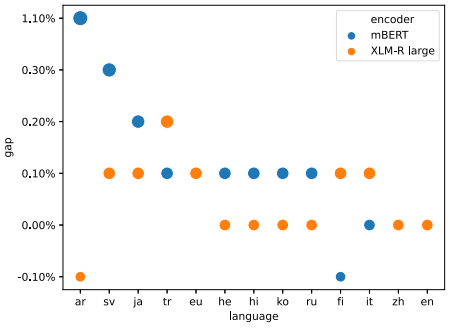
In this section, we will analyze the effectiveness of SL-XDP using different mPLMs on UD v2.5 through ablation studies with WAACL and MaChAmp. Our model shows benefits in terms of language and language family; the implementation of different feature fusion operations improves training efficiency, and the WAACL algorithm assists our model in not overfitting during training.

MPLM sensitivity. To investigate the impact of the mPLM encoder on language processing, we compared the performance of mBERT to that of a larger mPLM, XLM-R large. It has a transformer dimension of 1024, compared to mBERT's 768. All baseline models are re-trained using the "*xlm-roberta-large*" checkpoint on UD v2.5 and present the result in Tables 3 and 4 under the "XLM-R large" category. Our results show that XLM-R large generally performs better, with an average LAS of almost 90% for HRL and over 40% for LRL. Meanwhile, mBERT shows the greatest improvement with SL-XDP on both HRL and LRL, with general enhancements of over 0.2% on HRL and 0.1% on LRL. Due to the larger feature vector dimension in XLM-R, it may contain more complex information for SL-XDP to cipher or the misrepresentation of dependency knowledge on unseen language families.

Languages perspective. Following Üstün et al. [41], we analyze our results from a language perspective, taking into account the different situations of various languages. The LAS changes of SL-XDP and WAACL using the two mPLMs as backbones are presented in Figure 5. In terms of LRL, the improvements are not as significant, but mBERT can still achieve positive change since the positive range is extensive, while the negative scores are intensive, to zero. The most significant developments were found in AM with a 2.1% increase and MDF with a 1.4% increase. XLM-R showed the most improvement in MR and SA. Notably, these improved languages have a substantial difference in length distribution compared to HRL. On average, the length of HRL is around 18 words, while AM, MR, MDF, and SA sentences are all less than 10 words. This suggests that our model can capture the problems caused by unbalanced length distribution and mitigate their impact. As for HRL, we found that SL-XDP can improve mBERT's generalization more than XLM-R

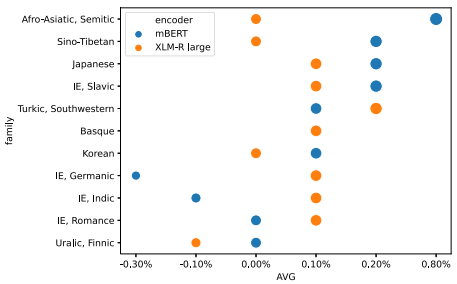


(a) Difference in LAS between SL-XDP and WAACL on LRL.

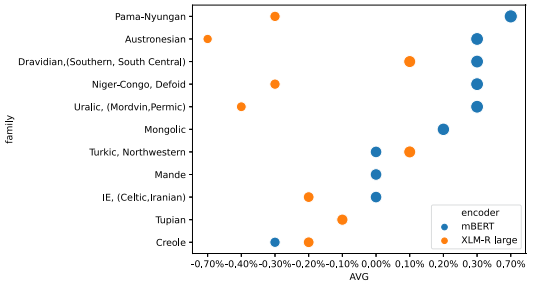


(b) Difference in LAS between SL-XDP and WAACL on HRL.

Fig. 5. The difference in LAS between SL-XDP and WAACL from the language perspective. The colors of the nodes indicate different PLM encoders. The languages are in code form for simplicity. For language code information, please refer to Appendix B.1.



(a) The average LAS difference between SL-XDP and WAACL for "*seen*" language familes.

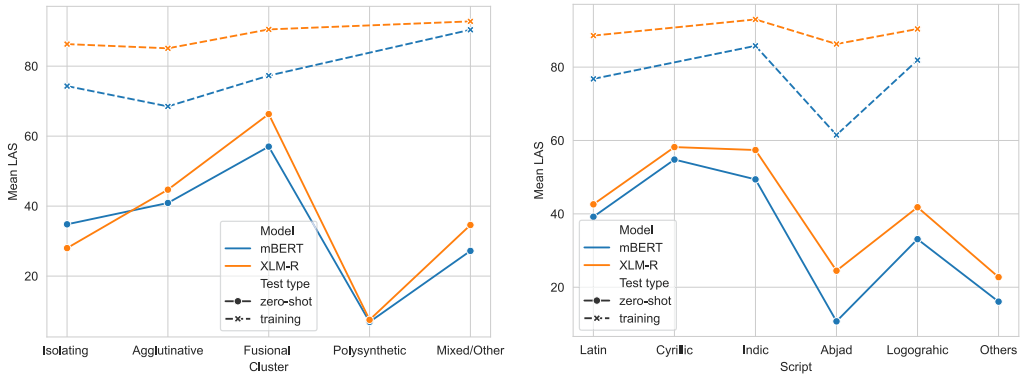


(b) The average LAS difference between SL-XDP and WAACL for "*unseen*" language familes.

Fig. 6. The average difference in LAS between SL-XDP and WAACL from the language family perspective. The colors of the nodes indicate different PLM encoders. For more information about language families, please refer to Appendix B.1.

large. Most of the gaps for mBERT are positive, with AR showing the highest gain by +1.1% and FI displaying the largest descent by -0.1%, resulting in a 1% difference.

Language families perspective. Model transfer for XDP relies on the similarity between languages [41]. With the typological features considered while selecting the training languages [41], we take another perspective into account: language families. Our analysis of the results, as shown in Figure 6, is based on whether a language family was "seen" or "unseen" during fine-tuning. The overall improvement for "seen" families was 1.6% for mBERT and 0.6% for XLM-R large. The Semitic family within the Afro-Asiatic language family, which includes AR, HE, AKK, AM, and AII languages, showed the highest improvement at 0.8%. For "unseen" languages, mBERT improved by 1.6%, while XLM-R descended by the same percentage. The most significant improvement for XLM-R was in the Pama-Nyungan family, which includes the WBP language at 0.7%. Our findings suggest that SL-XDP has a greater impact on seen language families for both mPLMs. However, the effect on unseen language families is less consistent, possibly because SL-XDP learns sample weights on seen language families but lacks knowledge of the dependency structures of unseen language families.



(a) Model performance on different morphological typology clusters.

(b) Model performance on different scripts.

Fig. 7. Mean LAS on different linguistic clusters. Colors indicate different encoders, and markers represent performance on training languages (x) and zero-shot languages (o).

Model behavior on different linguistic phenomenon. The experimental results also highlight several key points about the performance and behavior of mBERT and XLM-R when classifying languages by morphological typology [5] and scripts [7] (typology details are in Appendix B.2).

Behavior on different morphological typology clusters: Results in Figure 7(a) demonstrate that the performance of multilingual models is heavily influenced by morphological typology and the representation of languages in pre-training data. In zero-shot settings, both models struggle with isolating languages (around 30 for both models) due to their minimal morphology, while XLM-R outperforms mBERT on agglutinative languages (44.7 vs. 40.9), suggesting better handling of complex morphology. Both models perform well on fusional languages, with XLM-R excelling (66.3 vs. 57), likely due to their dominance in pre-training data. However, polysynthetic languages pose significant challenges, with both models performing poorly (under 10 for both models), indicating strong negative transfer effects from underrepresented morphological structures. Mixed/other languages show moderate performance, with XLM-R having a slight edge (34.6 vs. 27.2).

Training significantly improves performance across typologies, with XLM-R consistently outperforming mBERT. However, neither model overcomes the challenges of polysynthetic languages, achieving a LAS score of 0, and highlighting the need for specialized approaches. Mixed/other languages see near-perfect performance (92.8 for XLM-R, 90.4 for mBERT), suggesting training mitigates most negative transfer effects. Overall, the results underscore the influence of morphological typology and pre-training data representation on model performance, with polysynthetic languages revealing critical gaps in cross-linguistic generalization.

Behavior on different scripts: The experimental results in Figure 7(b) reveal the models' behavior in handling scripts with varying degrees of representation in pre-training data. In zero-shot settings, both XLM-R and mBERT demonstrate relatively strong performance on Latin and Cyrillic scripts, with XLM-R slightly outperforming mBERT. This suggests that these scripts, commonly associated with HRLs, are well-represented in the pre-training corpora of both models. However, performance drops significantly for scripts like Abjad and Logographic, particularly for mBERT (e.g., 10.7 for Abjad and 33.1 for Logographic), highlighting the challenges of generalizing to underrepresented scripts. XLM-R shows better zero-shot generalization (24.5 for Abjad, 41.8 for Logographic), likely due to its more diverse pre-training data. Notably, the "Others" category exhibits substantial negative transfer effects, especially for mBERT (16.075 vs. 22.75 for XLM-R), underscoring the limitations of cross-lingual transfer for scripts with minimal pre-training representation.

Table 5. Labeled Attachment Scores (LAS,%) on HRL of CoNLL Dataset

Methods	ar	zh	en	fi	he	hi	it	ja	ko	ru	sv	tr	AVG
MaChAmp	64.02%	63.11%	87.75%	78.08%	58.46%	50.80%	88.17%	82.23%	78.21%	81.37%	87.79%	37.90%	71.49%
WAACL	64.13%	62.42%	87.11%	77.49%	58.00%	50.69%	88.12%	82.34%	77.88%	81.95%	87.18%	37.80%	71.26%
SL-XDP (Ours)	**64.23%**	62.42%	87.13%	77.49%	58.02%	50.79%	88.05%	82.30%	77.95%	81.95%	87.29%	37.79%	71.29%

"AVG" shows the average LAS of HRL datasets.

Table 6. Labeled Attachment Scores (LAS,%) on LRL of CoNLL Dataset

Methods	bxr	hsb	kk	kmr	AVG
MaChAmp	25.72%	57.68%	54.94%	44.91%	45.81%
WAACL	27.16%	60.25%	53.48%	44.84%	46.43%
SL-XDP (Ours)	**27.41**%	**60.39**%	53.64%	44.50%	**46.49**%

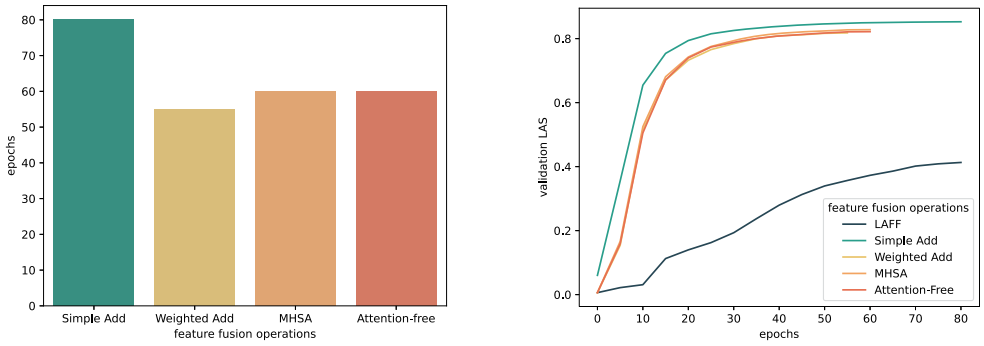
"AVG" shows the average LAS of LRL datasets.

When fine-tuned on specific scripts, both models exhibit improved performance, though with notable differences. Latin and Indic scripts yield strong results for both models, with XLM-R achieving particularly high scores (e.g., 88.6 for Latin and 93.0 for Indic). For Abjad and Logographic scripts, training significantly enhances performance, but XLM-R maintains a clear advantage (e.g., 86.3 vs. 61.5 for Logographic), demonstrating its superior adaptability to challenging scripts. These findings highlight the importance of pre-training data diversity and model architecture in mitigating negative transfer effects, particularly for underrepresented scripts. Overall, XLM-R's consistent outperformance across most script categories suggests it is better equipped to handle linguistic diversity, though both models struggle with scripts that are poorly represented in their pre-training corpora.

Potential negative transfer. Comparing the model performance between training languages and zero-shot languages in different clusters, we notice the negative transfer effects across various linguistic typologies, particularly in abjad and logographic scripts, as well as in polysynthetic languages. For abjad and logographic scripts, both models, especially mBERT, exhibit severe performance degradation during zero-shot testing. This suggests that the pre-trained embeddings of these models fail to adequately capture the unique linguistic features inherent to these scripts, such as the absence of vowels in abjads or the semantic complexity of logographic characters. Similarly, in polysynthetic languages, both models struggle to perform well in both zero-shot and fine-tuned settings. This indicates that the pretraining process does not sufficiently account for the morphological complexity and agglutinative nature of polysynthetic languages, leading to poor generalization and adaptation.

Furthermore, the "Others" and Mixed typology categories also demonstrate relatively poor zero-shot performance, likely due to the high linguistic diversity within these groups. The heterogeneity of languages in these categories appears to introduce interference effects, where the models' pretraining on more common or well-represented languages fails to transfer effectively to less common or typologically distinct languages. This interference underscores the challenges of creating multilingual models that can generalize across a wide range of linguistic typologies without suffering from negative transfer.

Generalization on different data type. We also conducted zero-shot experiments using XLM-R with the same language settings on the CoNLL 2017 dataset [47]. The testing data of CoNLL contains the 2.0 version UD data and new Parallel UD treebanks, which is different from UD v2.5. Results in Tables 5 and 6 suggest SL-XDP demonstrates strong generalization, excelling on zero-shot languages like BXR and HSB while maintaining competitive performance on training languages. It



(a) (b)

Fig. 8. (a) The training epoch for different feature fusion operations needs to reach optimal performance. (b) The validation of LAS during training for simple add and LAFF.

consistently matches or outperforms WAACL and MaChAmp, particularly on zero-shot languages. The deep stable learning framework mitigates distributional shifts, ensuring robust feature representation, while the feature fusion module enhances adaptability across linguistic structures. These findings show that SL-XDP is a promising framework for XDP, offering robust accuracy on training languages and improved performance on zero-shot languages, addressing the challenges of linguistic diversity and OOD scenarios.

Feature fusion improvement. Demonstrations of the effectiveness of five feature fusion operations use the mBERT as the mPLM encoder. Our findings indicate that Simple Add is the most effective fusion technique for both HRL and LRL. Most other fusion methods assist the model in achieving the optimal state with fewer training epochs than simple add, except LAFF, which we will explain in the end. Figure 8(a) illustrates that the Weighted add method takes only 55 epochs to be the earliest solution when the average testing LAS exceeds 80% and 36% for HRL and LRL, respectively. This suggests that assigning varying importance to tokens aids in generating a more comprehensive representation.

These fusion operations can introduce more improved languages. As mentioned above, simple add has been able to improve 13 out of 30 languages for LRL, while weighted add, MHSA, and Attention-Free have improved 14, 16, and 16 languages, respectively. However, it should be noted that these methods may not maintain the performance of all non-improved languages. Simple add has prevented a performance drop on 4 LRL, with all decreases being under 1%. Other operations have only been able to maintain the performance of 2 or 3 LRL languages, with the greatest decline being more than 6%. We argue that these feature fusion operations might filter the parsing knowledge during training and make the model unable to identify the parsing pattern in LRL.

Finally, the average LAS on both HRL and LRL is under 30% with LAFF fusion operation. After investigating the validation LAS during training in Figure 8(b), we observed that the validation LAS increases at a slower rate compared to other operations. Our theory is that converting the token representation to weights to generate the sentence representation alters the information in the vector space, which ultimately distracts from the training objectives.

Ablation study. Our SL-XDP model uses the WAACL algorithm to train dynamically with Worst-case Aware Automated Curriculum Learning. However, since the model does not observe all data points during a single training epoch, the sample weight SL-XDP learned may not be globalized. To test the impact of WAACL on SL-XDP, a variation called "-w/oWAACL" is created by removing

Table 7. The Average LAS Drop on Unseen Languages Using XLM-R Large

Language family	AVG LAS	AVG Length
IE, (Celtic,Iranian)	−0.2%	14.0
Uralic, (Mordvin,Permic)	−0.4%	9.1
Tupian	−0.1%	12.3
Creole	−0.2%	13.6
Austronesian	−0.7%	5.3
Pama-Nyungan	−0.3%	5.6
Niger-Congo, Defoid	−0.3%	26.6

"AVG LAS" denotes the average LAS drop among languages in the family."AVG Length" is the average length of the family. Families in the parentheses are a subclass of families for the family outside the parentheses.

the algorithm. The results show that without WAACL, the average LAS on HRL is generally higher at over 85%, but the performance on LRL decreases significantly. The original SL-XDP model has better LRL, with mBERT surpassing 36% and XLM-R large approaching 42% with more improved languages. SL-XDP improved mBERT by 6 and 4 languages in HRL and LRL, respectively. XLM-R large gained two more languages in both HRL and LRL with our original SL-XDP. We believe that both mPLM backbones overfit on the HRL datasets without the WAACL algorithm, which hinders the SL-XDP's ability to regulate the OOD impact and further improve. For more details, please refer to Tables 3 and 4.

Errors. Our SL-XDP has limited effectiveness in improving larger mPLMs, with 7 out of 11 language families experiencing a total loss of 2.1%, while the total improvement is only 0.5%. We suspect that this could be due to two reasons: (1) The large feature vectors contain too much information for SL-XDP to comprehend, requiring a more intricate information filtering implementation that can extract parsing information while removing the irrelevant noise. (2) SL-XDP may not have sufficient knowledge of how unseen language families construct their sentences, leading to misrepresentations. These families are listed in Table 7 for further investigation. Additionally, SL-XDP only enhances models that use the WAACL algorithm, resulting in a 0.2% improvement on HRL and 0.1% improvement on LRL. We believe that SL-XDP is not effective when models are overfitting with large datasets, and well-designed optimization strategies are necessary for assistance.

6 Discussion

DP is a fundamental task for natural language understanding, which is proven useful for human–machine interactions. Our research explores the OOD problem that arises due to the imbalanced length distribution among languages in the UD dataset and proposes the SL-XDP model, containing deep stable learning with a feature fusion module that mitigates the problem, resulting in improved model generalization across 43 languages. We argue that our research can offer five practical and theoretical implications.

6.1 Practical Implications

Based on our understanding of DP applications, we highlight two practical implications of SL-XDP to enhance AI systems.

First, the SL-XDP model can be useful by enhancing the natural language understanding abilities of information extraction systems. These systems rely on models to sift through extensive corpora and extract vital information that aligns with specific interests [50]. To accomplish this, models must be able to comprehend sentence semantics and recognize linguistic phenomena. The SL-XDP model, with its superior generalization, provides valuable guidance and enables the model to understand diverse languages. As a result, it could elevate extraction accuracy for event relation extraction in the general area [51] and biomedical entity recognition [46] in the medical domain.

In addition, the SL-XDP model can effectively handle real-world data by mitigating the impact of the OOD problem. Real-world data is often OOD and poses difficulties for models to comprehend. For instance, the PBSD model [45] that detects suicidality based on DP parsers for grammar understanding and false identification achieves only 68% accuracy on real-world data from Hong Kong. We contend that the performance plateau of PBSD is due to the OOD problem presented by real-world data, and suggest that the SL-XDP model can provide valuable assistance in reducing false alarms.

6.2 Theoretical Implications

By comparing with recent studies, our theoretical implications are threefold as follows.

In comparison with previous works of DP [2, 3, 14, 36], our study investigated a wider range of languages across 22 families and tested the LRL in the zero-shot setting, which poses a greater challenge for models. Our findings shed light on cross-lingual transferring, with our language-based results aligning with existing typology and linguistic studies. Specifically, we found that languages with greater typological [19] and linguistic [14] similarities tend to perform similarly on our model. For example, our SL-XDP LAS for RU and BE, which have higher lexical similarity,[3] outperformed previous models, while the LAS for FI and GSW, which are both "SVO" languages,[4] were lower. However, our result didn't completely support studies suggesting that language families can contribute to cross-lingual transferring [36]. The impact of language families varies depending on the scale of model parameters. Notably, we found that mBERT showed improvements on a large number of unseen language families, while XLM-R's performance dropped on most of these families.

Furthermore, introducing feature fusion modules has been proven effective for multi-modal studies [11, 18, 23]. Our SL-XDP suggests that the feature fusion methods are also efficient for single-modal studies. We have found that "add" operations, including "simple add" and "weighted add," have better generalization capacities for improving language while maintaining a small number of declines. Unlike multi-modal studies, we argue that keeping information intact is important for single-modal fusion. While LAFF [18] has been effective in reducing trainable parameters in text-to-video retrieval, it fails to reach optimal performance in SL-XDP. We believe this is because LAFF changes the information of the text features, thus distracting from the training objectives.

Finally, by integrating deep stable learning with a feature fusion module and a graph-based DP parser, our SL-XDP has paved the path to advancing the deep stable learning theory into the field of natural language understanding. Additionally, there are ongoing efforts to improve embedding representations for transition-based parser [52], develop semi-autoregressive parser [24], and improve high-order PD parser [26]. We hope that the success of the SL-XDP will inspire further research into these models, enabling us to tackle even more complex applications.

[3] https://www.ezglot.com/most-similar-languages?l=fin#most-similar-languages
[4] https://wals.info/feature/81A#2/18.0/153.1

7 Conclusion and Future Work

In our article, we outline the OOD problem as the difference in dependency structures between HRL and LRL, which causes the model to struggle in generalizing well on LRL datasets. Our research has introduced a new XDP model called SL-XDP to address the issues caused by the OOD problem. We've incorporated deep stable learning to eliminate complex relationships between random variables in the feature vector space. By learning sample weights, our SL-XDP can capture the distribution shift of the dependency structures of the vector space encoded by PLMs. Additionally, we've implemented five operations as a feature fusion procedure to generate a feature vector that contains XDP information for the deep stable learning algorithm. Our experiments on Universal Dependencies validate that SL-XDP enhances parsing abilities across a greater range of languages and even shows significant improvements in certain languages and language families.

Limitations. We discuss the limitations of our study and corresponding future works below.

Impact of Morphological Complexity: Many languages exhibit rich morphological variations that may interact with DP challenges. Future work could explore how SL-XDP handles morphological complexity and whether additional adjustments are needed.

Interpretability of deep stable learning: The deep stable learning algorithm is a key component of the approach, but its interpretability and explainability for linguistic tasks are not as visible as in computer vision tasks. Future work could focus on making the model's decisions more interpretable.

Resource Limitations in Low-Resource Languages: Even though our study made efforts to help model more generalized on more languages, XDP often involves LRLs that lack sufficient labeled data. Future research could examine how SL-XDP performs with minimal resources and explore methods to enhance performance in such scenarios.

Acknowledgments

Gratitude to the anonymous reviewers and the associate editor for their insightful comments.

References

[1] Kaushik Acharya. 2020. WNUT 2020 shared task-1: Conditional Random Field(CRF) based Named Entity Recognition(NER) for wet lab protocols. In *Proceedings of the 6th Workshop on Noisy User-Generated Text (W-NUT'20)*. Association for Computational Linguistics, Online, 286–289. DOI: https://doi.org/10.18653/v1/2020.wnut-1.37

[2] Mücahit Altıntaş and A. Cüneyd Tantuğ. 2023. Improving the performance of graph based dependency parsing by guiding bi-affine layer with augmented global and local features. *Intelligent Systems with Applications* 18 (2023), 200190. https://doi.org/10.1016/j.iswa.2023.200190

[3] Afra Amini, Tianyu Liu, and Ryan Cotterell. 2023. Hexatagging: Projective dependency parsing as tagging. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Toronto, Canada, 1453–1464.

[4] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Nan Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher J. Pal. 2020. A meta-transfer objective for learning to disentangle causal mechanisms. In *Proceedings of the 8th International Conference on Learning Representations*. openreview.net, Addis Ababa, Ethiopia.

[5] David W. Carroll. 1986. *Psychology of Language*. Thomson Brooks/Cole Publishing Co.

[6] Yoeng-jin Chu and Tseng-hong Liu. 1965. On the shortest arborescence of a directed graph. *Sci. Sinica* 14 (1965), 1396–1400.

[7] John Clews. 1984. World scripts (background paper). In *Proceedings of Translating and the Computer 6: Translation and Communication*. Catriona Picken (Ed.), Aslib, London, UK. Retrieved from https://aclanthology.org/1984.tc-1.15/

[8] Ayan Das and Sudeshna Sarkar. 2020. A survey of the model transfer approaches to cross-lingual dependency parsing. *ACM Transactions on Asian and Low-Resource Language Information Processing* 19, 5 (2020), 67:1–67:60.

[9] Miryam de Lhoneux, Sheng Zhang, and Anders Søgaard. 2022. Zero-shot dependency parsing with worst-case aware automated curriculum learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Dublin, Ireland, 578–587.

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

[11] Xue Dong, Xuemeng Song, Minghui Tian, and Linmei Hu. 2024. Prompt-based and weak-modality enhanced multi-modal recommendation. *Information Fusion* 101 (2024), 101989. https://doi.org/10.1016/j.inffus.2023.101989

[12] Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of the 5th International Conference on Learning Representations*. OpenReview.net, Toulon, France.

[13] Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards B* 71, 4 (1967), 233–240.

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

[15] Jeremy Barnes, Robin Kurtz, Stephan Oepen, Lilja Øvrelid, and Erik Velldal. 2021. Structured sentiment analysis as dependency graph parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 3387–3402. DOI: https://doi.org/10.18653/v1/2021.acl-long.263

[16] Nikolaos Giarelis and Nikos I. Karacapilidis. 2023. LMRank: Utilizing pre-trained language models and dependency parsing for keyphrase extraction. *IEEE Access* 11 (2023), 71459–71471. https://doi.org/10.1109/ACCESS.2023.3294716

[17] Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, 1234–1244.

[18] Fan Hu, Aozhu Chen, Ziyue Wang, Fangming Zhou, Jianfeng Dong, and Xirong Li. 2022. Lightweight attentional feature fusion: A new baseline for text-to-video retrieval. In *Proceedings of the 17th European Conference of Computer Vision* . Lecture Notes in Computer Science, Vol. 13674, Springer, 444–461.

[19] Tao Ji, Yuanbin Wu, and Xiaoling Wang. 2023. Typology guided multilingual position representations: Case on dependency parsing. In *Findings of the Association for Computational Linguistics: Association for Computational Linguistics 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 13524–13541.

[20] Kun Kuang, Peng Cui, Susan Athey, Ruoxuan Xiong, and Bo Li. 2018. Stable prediction across unknown environments. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1617–1626.

[21] Kun Kuang, Ruoxuan Xiong, Peng Cui, Susan Athey, and Bo Li. 2020. Stable prediction with model misspecification and agnostic distribution shift. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020, The 32nd Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The 10th AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, New York, NY, USA, 4485–4492.

[22] 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 the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers*. Association for Computational Linguistics, Dublin, Ireland, 8503–8520.

[23] Yi Lin, Dongyue Guo, Yuankai Wu, Lishuai Li, Edmond Q. Wu, and Wenyi Ge. 2024. Fuel consumption prediction for pre-departure flights using attention-based multi-modal fusion. *Inf. Fusion* 101, C (January 2024). DOI: https://doi.org/10.1016/j.inffus.2023.101983

[24] Ye Ma, Mingming Sun, and Ping Li. 2023. A semi-autoregressive graph generative model for dependency graph parsing. In *Findings of the Association for Computational Linguistics: Association for Computational Linguistics 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 4218–4230.

[25] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis M. Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 4034–4043.

[26] Farshad Noravesh. 2023. Transferring neural potentials for high order dependency parsing. arXiv:2306.10469. Retrieved from https://arxiv.org/abs/2306.10469

[27] Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, online, 7654–7673.

[28] Yinghao Piao and Jin-Xi Zhang. 2024. Text triplet extraction algorithm with fused graph neural networks and improved biaffine attention mechanism. *Applied Sciences* 14, 8 (2024), 3524. DOI: https://doi.org/10.3390/app14083524

[29] Esther Ploeger, Wessel Poelman, Miryam de Lhoneux, and Johannes Bjerva. 2024. What is "Typological Diversity" in NLP?. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Miami, Florida, USA, 5681–5700. DOI: https://doi.org/10.18653/v1/2024.emnlp-main.326

[30] Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 151–164. DOI: https://doi.org/10.18653/v1/P19-1015

[31] Ali Rahimi and Benjamin Recht. 2007. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20, Proceedings of the 21st Annual Conference on Neural Information Processing Systems*. Curran Associates, Inc., Vancouver, British Columbia, Canada, 1177–1184.

[32] Pooja Rai and Sanjay Chatterji. 2023. Annotation projection-based dependency parser development for Nepali. *ACM Transactions on Asian and Low-Resource Language Information Processing* 22, 2 (2023), 36:1–36:19.

[33] Zheyan Shen, Peng Cui, Kun Kuang, Bo Li, and Peixuan Chen. 2018. Causally regularized learning with agnostic data selection bias. In *Proceedings of the 2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 411–419.

[34] Zheyan Shen, Peng Cui, Jiashuo Liu, Tong Zhang, Bo Li, and Zhitang Chen. 2020. Stable learning via differentiated variable decomputer research repositoryelation. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, online, 2185–2193.

[35] Zheyan Shen, Peng Cui, Tong Zhang, and Kun Kuang. 2020. Stable learning via sample reweighting. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020, The 32th Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The 10th AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, New York, NY, USA, 5692–5699.

[36] 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* 45, 11 (2023), 13393–13407.

[37] Umar Taufiq, Reza Pulungan, and Yohanes Suyanto. 2023. Named entity recognition and dependency parsing for better concept extraction in summary obfuscation detection. *Expert Systems with Applications* 217 (2023), 119579. https://doi.org/10.1016/j.eswa.2023.119579

[38] Jörg Tiedemann and Željko Agic. 2016. Synthetic treebanking for cross-lingual dependency parsing. *J. Artif. Int. Res.* 55, 1 (January 2016), 209–248.

[39] Jörg Tiedemann, Zeljko Agic, and Joakim Nivre. 2014. Treebank translation for cross-lingual parser induction. In *Proceedings of the 18th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Baltimore, Maryland, USA, 130–140.

[40] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. 2019. An empirical study of example forgetting during deep neural network learning. In *Proceedings of the 7th International Conference on Learning Representations*. OpenReview.net, New Orleans, LA, USA.

[41] Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly universal dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online, 2302–2315.

[42] Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, European Chapter of the Association for Computational Linguistics, Online*. 176–197.

[43] Shijie Wu and Mark Dredze. 2019. Beto, Bentz, Becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, Hong Kong, China, 833–844.

[44] Min Xiao and Yuhong Guo. 2014. Distributed word representation learning for cross-lingual dependency parsing. In *Proceedings of the 18th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Baltimore, Maryland, USA, 119–129.

[45] Zhongzhi Xu, Christian S. Chan, Jerry Fung, Christy Tsang, Qingpeng Zhang, Yucan Xu, Florence Cheung, Weibin Cheng, Evangeline Chan, and Paul S. F. Yip. 2023. Developing and validating a parser-based suicidality detection model in text-based mental health services. *Journal of Affective Disorders* 335 (2023), 228–232. https://doi.org/10.1016/j.jad.2023.04.128

[46] Yumeng Yang, Hongfei Lin, Zhihao Yang, Yijia Zhang, Di Zhao, and Shuaiheng Huai. 2023. ADPG: Biomedical entity recognition based on automatic dependency parsing graph. *Journal of Biomedical Informatics* 140 (2023), 104317. https://doi.org/10.1016/j.jbi.2023.104317

[47] Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the*

CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. Association for Computational Linguistics, Vancouver, Canada, 1–19. DOI: https://doi.org/10.18653/v1/K17-3001

[48] Maoyuan Zhang, Lisha Liu, Jiaxin Mi, and Xianqi Yuan. 2023. Enhanced dual-level dependency parsing for aspect-based sentiment analysis. *The Journal of Supercomputing* 79, 6 (2023), 6290–6308.

[49] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. 2021. Deep stable learning for out-of-distribution generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation / IEEE, online, 5372–5382.

[50] Shaowen Zhou, Bowen Yu, Aixin Sun, Cheng Long, Jingyang Li, and Jian Sun. 2022. A survey on neural open information extraction: Current status and future directions. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*. ijcai.org, Vienna, Austria, 5694–5701.

[51] Ling Zhuang, Hao Fei, and Po Hu. 2023. Syntax-based dynamic latent graph for event relation extraction. *Information Processing and Management* 60, 5 (2023), 103469.

[52] Fatima Tuz Zuhra and Khalid Saleem. 2023. Hybrid embeddings for transition-based dependency parsing of free word order languages. *Information Processing and Management* 60, 3 (2023), 103334.

Appendices

A Explanation of Deep Stable Learning in SL-XDP

In this section, we provide pieces of evidence for deep stable learning in the DP task and mitigate the potential limitations of the biaffine network.

A.1 Challenges in the Biaffine Network for Cross-Lingual Dependency Parsing

The biaffine network, while effective for DP, is limited by two major challenges [28].

Shallow Contextual Understanding. : Biaffine networks primarily rely on token-level interactions to predict syntactic dependencies but may struggle to capture deeper hierarchical relationships in complex sentences. In cases of deeply nested or ambiguous syntactic structures, the network's limited contextual scope can lead to errors, particularly in OOD scenarios.

Static Dependency Parsing Assumptions. : Biaffine networks assume that syntactic structures are relatively fixed across languages. However, natural language is inherently dynamic, and the syntactic relations in one language may not map directly to another. This issue is exacerbated by the unbalanced sentence-length distributions between training and testing languages, where the model often overfits the length patterns in the training data.

These limitations reduce the biaffine network's ability to generalize across languages, particularly when faced with structural diversity and OOD problems.

A.2 Role of Deep Stable Learning in SL-XDP

Deep stable learning in SL-XDP is designed to mitigate these challenges through a series of methodological enhancements, ensuring the model can generalize effectively across diverse languages and sentence structures.

Enhanced Contextual Representation through Feature Fusion. To address the biaffine network's limited contextual understanding, SL-XDP integrates a Feature Fusion Module with five fusion operations (Equations (5)–(14)).

Multi-head Self-Attention and LAFF: These operations enrich the token representations by capturing long-range dependencies and hierarchical relationships. By generating more comprehensive sentence representations, the model can better handle deeply nested or ambiguous syntactic structures.

Simple Add, Weighted Add, and Attention-Free Fusion: Fusion mechanisms like simple add and weighted addition allow the model to emphasize important tokens, improving the

precision of DP in complex sentences. Attention-free methods offer computational efficiency while maintaining high representational quality.

By combining these fusion strategies, SL-XDP enhances the biaffine network's ability to process contextual and hierarchical information, which is critical for resolving ambiguities in syntactic relationships.

Decorrelation of Dependency Structures from Sequence Length. One key aspect of the OOD problem in XDP is the unbalanced sentence-length distribution among languages. Deep stable learning directly addresses this by decoupling dependency predictions from sequence length biases.

Random Fourier Feature (RFF) Augmentation: As presented in Equation (16), augments feature representations by introducing RFF, which help the model focus on the underlying dependency relations rather than surface-level biases like sequence length. This ensures that the model can generalize dependency structures across languages with varying sentence lengths.

Decorrelation Weight Learning: Like Equation (21), deep stable learning iteratively learns weights for each training sample to reduce the correlation between dependency predictions and sequence length. This prevents the model from overfitting to spurious patterns associated with sentence length, thereby improving generalization to OOD languages.

A.3 Iterative Weight Updates for Robustness

Deep stable learning employs an iterative reweighting mechanism to improve the model's robustness to OOD scenarios.

Reweighting with deep stable learning. : As presented in Equation (21), during training, deep stable learning dynamically adjusts the weights of samples based on their contribution to the model's performance. This ensures that the model focuses on challenging or misaligned samples, improving its ability to adapt to OOD sentence structures.

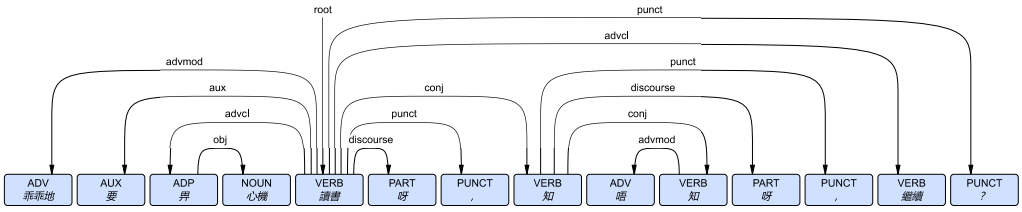
Saving and Reloading Mechanism. : The iterative process of saving and reloading learned weights helps the model retain generalizable patterns while discarding spurious correlations. This mechanism is particularly effective in handling structural variability across languages.

A.4 Empirical Evidence

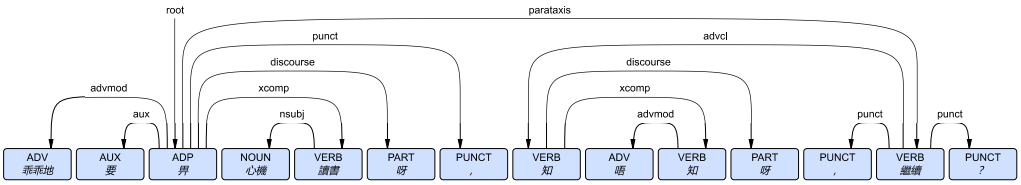
We also provide a case analysis in Figure 9 of SL-XDP's prediction against the gold truth on a Cantonese DP task in comparison to MaChAmp and WAACL. The results highlight SL-XDP's deep stable learning advantages in accurately capturing syntactic and semantic dependencies:

(1) **Accurate Core Dependencies**: SL-XDP reliably predicts dependency center words and essential relations like *advmod*, *aux*, and *obj*, ensuring accurate sentence structure representation, unlike other models that misassign key dependencies.
(2) **Robust Complex Structures**: It effectively handles challenging patterns, such as relations between the two components on the two sides of the first comma, showing resilience in learning hierarchical and non-projective structures.
(3) **Minimized Error Propagation**: Stable predictions of key dependencies reduce cascading errors, ensuring more reliable dependency trees.

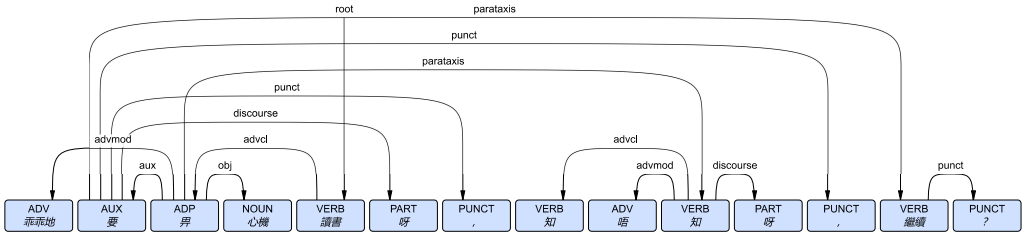
These advancements allow SL-XDP to overcome the OOD problem in XDP, improving the biaffine network's capacity to handle structural diversity and generalize across languages.



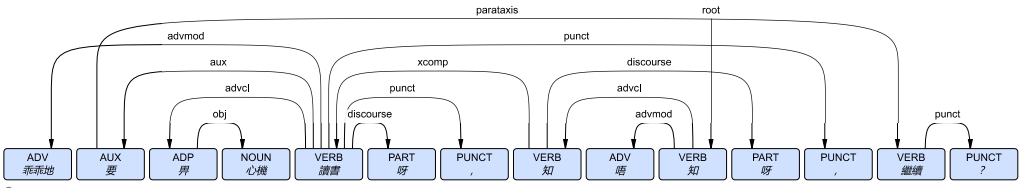
(a) Gold truth.



(b) MaChAmp prediction.



(c) WAACL prediction.



(d) SL-XDP prediction.

Fig. 9. A case of predictions on the Cantonese dataset. The sentence means "Be obedient to study hard, (you) know, (can we) continue?". SL-XDP predicts the most correct relations.

B Language Information

B.1 Language Details

Tables 8 and 9 provide information on training and zero-shot languages, including the language code, data size (number of sentences), treebanks, and family.

We also present the language analysis for typological diversity in terms of geographic location, and syntactic distances using the mean pairwise distance metric proposed by Ploeger et al. [29].

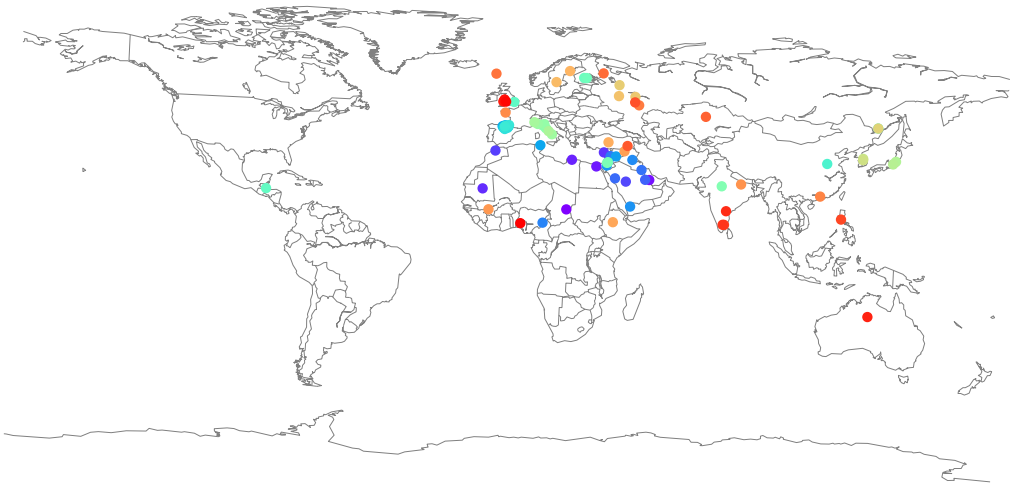
As presented in Tables 8 and 9, the selection of 43 languages for dependency analysis spans a wide geographic and linguistic diversity, representing multiple families such as Indo-European, Sino-Tibetan, and Afro-Asiatic, among others. This variety allows for the assessment of the model's

Table 8. Training Language Information

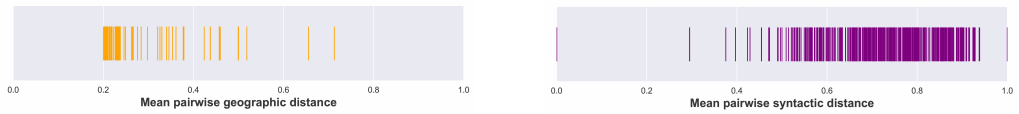
Language	Code	Treebank	Family
Arabic	ar	PADT	Afro-Asiatic, Semitic
Basque	eu	BDT	Basque
Chinese	zh	GSD	Sino-Tibetan
English	en	EWT	IE, Germanic
Finnish	fi	TDT	Uralic, Finnic
Hebrew	he	HTB	Afro-Asiatic, Semitic
Hindi	hi	HDTB	IE, Indie
Italian	it	ISDT	IE, Romance
Japanese	ja	GSD	Japanese
Korean	ko	GSD	Korean
Russian	ru	SynTagRus	IE, Slavic
Swedish	sv	Talbanken	IE, Germanic
Turkish	tr	IMST	Turkic. Southwestern

Table 9. Zero-Shot Languages Information

Language	Code	Treebank(s)	Family
Akkadian	akk	PISANDUB	Afro-Asiatic, Semitic
Amharic	am	ATT	Afro-Asiatic, Semitic
Assyrian	aii	AS	Afro-Asiatic, Semitic
Bambara	bm	CRB	Maude
Belarusian	be	HSE	IE, Slavic
Bhojpuri	bho	BHTB	IE, Indie
Breton	br	KEB	IE, Celtic
Buryat	bxr	BDT	Mongolic
Cantonese	yue	HK	Sino-Tibetan
Erzya	myv	JR	Uralic, Mordvin
Faroese	fo	OFT	IE, Germanic
Karelian	kr1	KKPP	Uralic, Finnic
Kazakh	kk	KTB	Turkic, Northwestern
Komi Permyak	koi	UH	Uralic, Permic
Komi Zyrian	kpv	LATTICE, IKDP	Uralic. Perniic
Kurmanji	kmr	MG	IE, Iranian
Livvi	olo	KKPP	Uralic, Finnic
Marathi	mr	UFAL	IE, Indic
Mbya Guarani	gun	THOMAS, DOOLEY	Tupian
Moksha	mdf	JR	Uralic. Mordvin
Naija	pcm	NSC	Creole
Sanskrit	sa	UFAL	IE, Indic
Swiss G.	gsw	UZH	IE, Germanic
Tagalog	t1	TRG	Austronesian, Central Philippine
Tamil	ta	TTB	Dravidian, Southern
Telugu	te	MTG	Dravidian, South Central
Upper Sorbian	hsb	UFAL	IE, Slavic
Warlpiri	wbp	UFAL	Pama-Nyungan
Welsh	cy	CCG	IE, Celtic
Yoruba	yo	YTB	Niger-Congo, Defoid



(a) Language geographic location distribution.



(b) Mean pairwise distance distribution.

Fig. 10. Language diversity analysis. Our chosen languages cover a wide range of syntactic features, a large language use population, and a rich linguistic history.

ability to adapt to diverse syntactic structures and generalize across different linguistic contexts (Figure 10). The balanced geographic distribution and inclusion of languages with varying syntactic and spatial distances ensure that the model is not biased toward specific regions or grammatical systems, strengthening its robustness and global applicability.

B.2 Language Classification

Furthermore, we present the classification of zero-shot languages based on Morphological Typology and Script for in-depth analysis of model behaviors on different linguistic phenomena and potential negative transfer effects.

Morphological Typology. Morphological typology [5] is a way of classifying languages based on how they use morphemes (the smallest units of meaning) to form words. It describes the structure of words and how languages encode grammatical information, such as tense, number, case, or gender. Languages can be broadly grouped into several categories based on their morphological characteristics:

— Isolating (Analytic) Languages: Words typically consist of a single morpheme, and grammatical relationships are expressed through word order and auxiliary words rather than inflection.
— Agglutinative Languages: Words are formed by stringing together multiple morphemes, each representing a single grammatical function (e.g., tense, case, number).

Table 10. Our Classification of Zero-Shot Languages According to Morphological Typology Features

Typology	Training Languages	Zero-shot Languages
Isolating	Chinese (zh)	Cantonese (yue), Naija (pcm), Yoruba (yo)
Agglutinative	Finnish (fi), Korean (ko), Turkish (tr)	Kazakh (kk), Komi Permyak (koi), Komi Zyrian (kpv), Livvi (olo), Tamil (ta), Telugu (te), Buryat (bxr)
Fusional	Arabic (ar), Basque (eu), English (en), Hebrew (he), Hindi (hi), Italian (it), Russian (ru), Swedish (sv)	Belarusian (be), Breton (br), Faroese (fo), Marathi (mr), Sanskrit (sa), Upper Sorbian (hsb)
Polysynthetic	–	Warlpiri (wbp)
Mixed/Other	Japanese (ja)	Akkadian (akk), Amharic (am), Assyrian (aii), Mbya Guarani (gun), Tagalog (tl), Bambara (bm), Bhojpuri (bho), Karelian (krl), Kurmanji (kmr), Moksha (mdf), Swiss G (gsw), Welsh (cy), Erzya (myv)

Table 11. Our Classification of Languages According to Language Scripts

Script Type	Training Languages	Zero-shot Languages
Latin	Basque (eu), English (en), Finnish (fi), Italian (it), Swedish (sv), Turkish (tr)	Breton (br), Faroese (fo), Komi Permyak (koi), Komi Zyrian (kpv), Livvi (olo), Mbya Guarani (gun), Naija (pcm), Tagalog (tl), Switzerland German (gsw), Upper Sorbian (hsb), Warlpiri (wbp), Welsh (cy), Yoruba (yo)
Cyrillic	–	Belarusian (be), Buryat (bxr), Kazakh (kk)
Indic	Hindi (hi)	Bhojpuri (bho), Hindi (hi), Marathi (mr), Sanskrit (sa), Tamil (ta), Telugu (te)
Abjad	Arabic (ar), Hebrew (he)	Amharic (am), Akkadian (akk), Assyrian (aii)
Logographic	Chinese (zh), Japanese (ja), Russian (ru), Korean (ko)	Cantonese (yue)
Others	–	Bambara (bm), Zrzya (myv), Kurmanji (kmr), Moksha (mdf)

— Fusional (Inflectional) Languages: Words use affixes that often encode multiple grammatical features simultaneously (e.g., tense and number in a single suffix).
— Polysynthetic Languages: Words are highly complex and can include multiple morphemes that represent entire sentences or phrases.
— Mixed Typology: Some languages combine features from multiple types, making them harder to classify strictly.

We classify experimental languages as Table 10.

Script. The script of a language refers to the system of writing used to represent its sounds, words, or concepts [7]. Scripts can vary widely in their structure and complexity, influencing how easily a language can be processed by models, especially in zero-shot or low-resource settings. Common types of scripts include:

— Alphabetic Scripts: These scripts use symbols (letters) to represent individual sounds (phonemes), like Latin and Cyrillic.
— Abjad (Consonant-Based Scripts): These scripts primarily represent consonants, and vowels are either optional or indicated using diacritics
— Abugida (Syllabic-Based Scripts): These scripts represent syllables, where each symbol typically combines a consonant and a vowel.
— Logographic Scripts: These scripts use symbols (characters) to represent entire words or concepts rather than sounds.

We classify experimental languages as Table 11.

C Zero-Shot Results

Table 12 shows the complete zero-shot results for UDapter, MaChAmp, WAACL, SL-XDP, and SL-XDP - w/oWAACL. Models with † represent results on UD v2.3. Some languages are not included

Table 12. Labeled Attachment Scores (LAS,%) on All 30 LRL for UDapter†, WAACL, SL-XDP, MaChAmp, and SL-XDP -w/oWAACL Using Two mPLM Backbones (mBERT and XML-R Large)

languages	mBERT						XLM-R large			
	UDapter†	WAACL	MaChAmp	SL-XDP†	SL-XDP	SL-XDP - w/oWAACL	WAACL	MaChAmp	SL-XDP	SL-XDP - w/oWAACL
aii	14.3	7.9	5.3	–	**8.4**	4	2.9	4	2.9	2.9
akk	8.2	1.5	1.9	1.3	1.6	1.9	2.1	2.4	2.2	1.9
am	5.9	20.0	4.1	**24.3**	**22.1**	**4.8**	68.4	67.8	68.4	67.0
be	79.3	78.1	81.5	77.5	78.5	81.4	85.3	86.5	85.3	86.2
bho	37.3	38.0	37.6	–	37.8	37.6	37.3	37.2	37.3	37.1
bm	8.1	9.0	6.9	7.7	9.0	6.7	6.2	6.2	6.2	6.0
br	58.5	62.8	61	**62.4**	62.8	61	59.3	60.0	59.0	59.9
bxr	28.9	25.8	24.6	26.1	**26.0**	**24.8**	27.3	25.8	**27.5**	25.8
cy	54.4	55.4	55	–	**55.5**	54.9	59.9	59.8	59.9	59.7
fo	69.2	67.6	66.5	66.5	67.0	**66.6**	72.5	72.3	**72.6**	**72.7**
gsw	45.5	49.3	43.3	–	48.4	**43.5**	44.9	43.4	**45.2**	40.9
gun	8.4	4.7	4.4	–	4.6	4.2	4.8	4.0	4.6	3.7
hsb	54.2	50.6	50.1	50.4	**50.7**	**50.3**	61.4	58.7	**61.5**	**60.2**
kk	60.7	59.8	58.3	59.6	59.8	58.3	61.7	63.1	61.8	62.7
kmr	12.1	9.3	8.8	8.9	9.1	8.5	52.0	51.9	51.7	51.9
koi	23.1	19.8	15.3	–	19.5	15	19.3	16.3	**19.5**	**16.5**
kpv	12.5	17.0	16.4	**16.6**	17.0	16.1	17.9	16.8	17.7	16.5
krl	48.4	46.2	46.5	–	46.1	46.1	61.9	60.9	**62.0**	**61.4**
mdf	26.6	24.5	24.1	–	**25.9**	24.1	18.4	18.8	17.0	17.7
mr	44.4	60.4	56.7	**59.0**	60.1	55.1	68.6	67.8	**68.9**	**68.6**
myv	19.2	20.4	18.7	**19.9**	20.3	18.7	16.1	16.0	16.1	15.4
olo	43.3	40.9	41.1	–	**41.1**	40.8	56.5	53.0	56.1	53.0
pcm	36.7	33.7	33.5	33.8	33.4	**34.3**	38.8	38.5	38.7	37.5
sa	22.2	23.3	20.4	21.6	23.0	20.3	50.3	51.9	**50.6**	48.9
ta	46.1	52.4	52.1	**52.3**	52.3	**52.3**	54.0	52.4	53.9	**52.7**
te	71.1	69.9	70.3	70.5	**70.6**	70.2	75.9	76.1	76.1	74.6
tl	69.5	65.1	65.1	63.7	**65.4**	65.1	76.4	77.7	75.7	75.7
wbp	12.1	6.2	6.5	6.9	**6.9**	5.9	7.8	11.8	7.5	**12.1**
yo	42.7	38.3	39.6	33.5	37.8	**39.7**	3.8	2.6	3.5	2.4
yue	32.8	32.8	31.7	**32.9**	**33.1**	**31.9**	41.8	43.1	41.8	42.4
AVG	36.5	36.4	34.9	**37.9**	**36.5**	34.8	41.8	41.6	41.7	41.1

"AVG" represents the average LAS score of all listed languages. The UDapter† and SL-XDP† are trained on UD v2.3, and we use the report of UDapter from their paper. Some languages are not included in the latest archive, which marked "–" as their results.

in the latest UD v2.3 archive,[5] so we have marked their results as "–". All languages were not seen during the model training process and were directly tested after finishing fine-tuning.

D Generalization Ability on NER Application

This section presents an examination of SL-XDP in the context of real-world applications, specifically focusing on **named entity recognition** (**NER**).

D.1 Experiment Setting

We train the XLM-R model using the multilingual NER dataset WikiANN [30]. Adopting a similar approach to XDP, we utilized 13 topologically diverse HRLs and 18 LRLs for both training and zero-shot testing languages. The extraction performance is evaluated using the Span F1 score, as defined in Equation (25).

$$F1 = \frac{2 * P * R}{P + R} \tag{25}$$

[5] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2895

where $P = \frac{TP}{TP+FP}$ and $R = \frac{TP}{TP+FN}$, TP =true positive, TP =true negative, FP =false positive, FN =false negative.

We evaluate the performance of our SL-XDP in comparison to the baseline methods: MaChAmp and WAACL. All models employ a **Conditional Random Field** (**CRF**) [1] decoder to address the NER task. We maintain consistent configurations for hyperparameters in the XDP task and train each model for 50 epochs.

D.2 Results

The results in Table 13 highlight the performance of SL-XDP and the baselines (MaChAmp, WAACL) in NER tasks across training languages and zero-shot languages.

Table 13. Span F1 Score (%) on 13 HRL for MaChAmp, WAACL, and SL-XDP using XLM-R Large Backbones

Methods	ar	bn	en	fi	he	hi	it	ja	ko	ru	sv	tr	zh	AVG
MaChAmp	28.32%	41.51%	45.05%	48.55%	26.70%	39.89%	54.55%	11.83%	28.05%	26.61%	61.38%	48.21%	23.51%	37.24%
WAACL	0.33%	0.34%	0.34%	0.65%	0.70%	0.84%	0.40%	0.15%	4.18%	0.42%	0.35%	0.67%	0.10%	0.73%
SL-XDP (Ours)	**90.67%**	**95.68%**	**84.96%**	**91.85%**	**87.04%**	**91.03%**	**92.01%**	**68.21%**	**88.05%**	**89.62%**	**95.06%**	**93.22%**	**77.17%**	**88.05%**

"AVG" represents the average F1 score of all listed languages.

Performance on HRL. SL-XDP achieves a mean Span F1 of 88.05% across training languages, significantly surpassing MaChAmp (37.24%) and WAACL (0.73%).

Specifically, SL-XDP demonstrates robust performance across typologically diverse languages, including HRLs (e.g., English: 84.96%) and LRLs (e.g., Arabic: 90.67%, Korean: 88.05%, Swedish: 95.06%). The substantial performance gap over WAACL and MaChAmp (Δ > 50 F1 points) underscores SL-XDP's effectiveness in mitigating distributional shifts inherent in multilingual data. The feature fusion module likely enables richer representation learning by integrating cross-lingual syntactic and semantic features. High F1 scores in both resource-rich (e.g., Swedish) and resource-constrained (e.g., Korean) languages indicate that deep stable learning reduces overfitting to dominant language features, promoting balanced parameter optimization.

The method's reliability in multilingual training environments makes it suitable for applications requiring consistent NER performance, such as multilingual content moderation or information extraction systems in global enterprises. Its stability across typologies ensures applicability to languages with scarce annotated resources, reducing dependency on extensive retraining.

Performance on LRL. SL-XDP achieves a mean zero-shot F1 of 71.38% (see Table 14), vastly surpassing MaChAmp (30.95%) and WAACL (1.83%), highlighting its cross-lingual transfer efficacy.

Particularly, SL-XDP attains notable F1 scores in languages absent from training data, such as Bengali (83.62%), Bhojpuri (69.17%), and Tagalog (82.49%). Even for challenging cases like Akan (53.85%) and Kurdish (56.28%), it outperforms baselines by >50 F1 points, demonstrating resilience to typological divergence. We argue that the feature fusion module likely enables the abstraction of language-agnostic patterns (e.g., entity boundaries, contextual dependencies), facilitating transfer to unseen languages. This contrasts with WAACL and MaChAmp, which fail to generalize due to rigid alignment strategies or over-reliance on language-specific embeddings. SL-XDP's success in low-resource zero-shot settings (e.g., Welsh: 72.78%) suggests its utility in scenarios where annotated data is unavailable, a common challenge in real-world deployments.

The method's zero-shot capability is critical for applications involving under-resourced languages, such as disaster response systems or customer support tools in multilingual regions. Its ability to generalize without retraining reduces operational costs and accelerates deployment in dynamic environments.

Table 14. Span F1 Score (%) on 18 LRL for MaChAmp, WAACL, and SL-XDP Using XLM-R Large Backbones

Languages	MaChAmp	WAACL	SL-XDP (Ours)
am	18.78%	2.97%	**53.85%**
be	48.14%	0.87%	**83.62%**
bho	41.63%	1.60%	**69.17%**
br	34.96%	0.49%	**71.68%**
cy	39.50%	0.20%	**72.78%**
fo	44.68%	2.17%	**84.77%**
gun	20.00%	1.51%	**69.26%**
hsb	43.60%	0.24%	**82.16%**
kk	29.84%	1.86%	**59.13%**
kmr	24.00%	7.78%	**56.28%**
mr	35.62%	0.59%	**83.71%**
myv	10.81%	4.00%	**75.83%**
sa	32.09%	2.15%	**62.41%**
ta	20.85%	2.14%	**80.40%**
te	18.17%	3.93%	**70.79%**
tl	42.18%	0.42%	**82.49%**
yo	25.74%	0.00%	**52.49%**
yue	26.55%	0.05%	**73.95%**
AVG	30.95%	1.83%	**71.38%**

"AVG" represents the average F1 score of all listed languages.

In conclusion, SL-XDP establishes a new method for cross-lingual NER, combining deep stable learning with feature fusion to achieve state-of-the-art performance on both training and zero-shot languages. Its strong generalization across linguistic and resource-based disparities highlights its potential for real-world multilingual NLP systems, particularly in scenarios requiring adaptability to data scarcity or typological diversity.

Received 26 March 2024; revised 26 February 2025; accepted 4 May 2025