Hyber-X: A Unified Hypernetwork for Multi-Task Multilingual Transfer

Ahmet Üstün1, Arianna Bisazza1, Gosse Bouma1, Gertjan van Noord1, Sebastian Ruder2

1University of Groningen
2Google Research
a.ustun@rug.nl

Abstract
Massively multilingual models are promising for transfer learning across tasks and languages. However, existing methods are unable to fully leverage training data when it is available in different task-language combinations. To exploit such heterogeneous supervision, we propose Hyper-X, a single hypernetwork that unifies multi-task and multilingual learning with efficient adaptation. This model generates weights for adapter modules conditioned on both tasks and language embeddings. By learning to combine task and language-specific knowledge, our model enables zero-shot transfer for unseen languages and task-language combinations. Our experiments on a diverse set of languages demonstrate that Hyper-X achieves the best or competitive gain when a mixture of multiple resources is available, while being on par with strong baselines in the standard scenario. Hyper-X is also considerably more efficient in terms of parameters and resources compared to methods that train separate adapters. Finally, Hyper-X consistently produces strong results in few-shot scenarios for new languages, showing the versatility of our approach beyond zero-shot transfer.

1 Introduction
Transfer learning across languages and tasks has long been an important focus in NLP (Ruder et al., 2019). Recent advances in massively multilingual transformers (MMTs; Devlin et al., 2019; Conneau et al., 2020) show great success in this area. A benefit of such models is their ability to transfer task-specific information in a high-resource source language to a low-resource target language (Figure 1, (1)). Alternatively, such models can leverage knowledge from multiple tasks for potentially stronger generalization (Figure 1, (2)).

Over time, many research communities have been developing resources for specific languages of focus (Strassel and Tracey, 2016; Nivre et al., 2018; Wilie et al., 2020). In practice, it is thus common for data to be available for different tasks in a mixture of different languages. For instance, in addition to English data for both POS tagging and Named Entity Recognition (NER), a treebank with POS annotation may be available for Turkish, while NER data may be available for Arabic. This example is illustrated in Figure 1, (3).

In contrast to existing cross-lingual transfer paradigms such as single-task zero-shot transfer (Hu et al., 2020) or few-shot learning (Lauscher et al., 2020a), multi-task learning on such a mixture of datasets (mixed-language multi-task) poses an opportunity to leverage all available data and
Table 1: A comparison of existing approaches and Hyper-X based on their transfer capabilities. We characterize approaches based on whether they can perform cross-lingual transfer (X-Lang.) and cross-task transfer via multi-task learning (M-Task) in the zero-shot setting or to unseen language-task pairs (X-Pair). As a particular case of cross-lingual transfer, ‘New Lang’ represents the case when transfer is generalizable to unseen languages not covered by the multilingual pre-trained model.

| Model                      | Description                                           | X-Lang. | New Lang. | M-Task | X-Pair (LT) |
|----------------------------|-------------------------------------------------------|---------|-----------|--------|-------------|
| MAD-X (Pfeiffer et al., 2020b) | Cross-lingual transfer via language/task adapters    | ✔       | ✔         | ✗      | ✔           |
| HyperFormer (Mahabadi et al., 2021b) | Multi-task learning via shared hypernet adapters     | ✗       | ✗         | ✔      | ✗           |
| Parameter Space Fact. (PSF; Ponti et al., 2021) | Transfer to unseen task-language pairs via PSF       | ✗       | ✗         | ✔      | ✔           |
| Hyper-X (this work)        | Multi-language/task transfer via a unified hypernet  | ✔       | ✔         | ✔      | ✔           |

To transfer information across both tasks and languages to unseen task–language combinations (Ponti et al., 2021).

Standard fine-tuning strategies, however, are limited in their ability to leverage such heterogeneous task and language data. Specifically, MMTs are prone to suffer from catastrophic forgetting and interference (Wang et al., 2020) when they are fine-tuned on multiple sources. Adapters (Houlsby et al., 2019), a parameter-efficient fine-tuning alternative are commonly used for transfer either across tasks (Mahabadi et al., 2021b) or languages (Üstün et al., 2020) but require training a new adapter for each new language (Pfeiffer et al., 2020b).

In this paper, we propose a unified hypernetwork, Hyper-X that is particularly suited to this setting by leveraging multiple sources of information including different languages and tasks within a single model. The core idea consists of taking language and task embeddings as input, and generating adapter parameters via a hypernetwork for the corresponding task-language combination. By parameterizing each task and language separately, Hyper-X enables adaptation to unseen combinations at test time while exploiting all available data resources.

Additionally, Hyper-X can make seamless use of masked language modelling (MLM) on unlabelled data, which enables it to perform zero-shot adaptation to languages not covered by the MMT during pre-training. MLM also enables Hyper-X to learn a language representation even without available task-specific data.

In sum, our work brings together a number of successful transfer ‘ingredients’ that have been explored in very recent literature (see Table 1), namely multi-task learning, multilingual learning, further pre-training, along a high degree of compute- and time-efficiency.

We evaluate Hyper-X for cross-lingual transfer on two sequence labelling tasks, namely part-of-speech (POS) tagging and named-entity recognition (NER) in 16 languages—7 of which are not covered in pre-training—across the three experimental setups depicted in Figure 1. Our experiments demonstrate that Hyper-X is on par with strong baselines for cross-lingual transfer from English. In the multi-task and mixed-language settings, Hyper-X shows a large improvement compared to the standard baselines and matches the performance of the less efficient adapter-based model due to its ability to leverage heterogeneous sources of supervision. Analysis highlights that Hyper-X is superior in terms of efficiency–performance trade-offs. Finally, we evaluate our model in a few-shot setting, where Hyper-X consistently achieves competitive performance across different languages and tasks, which suggests the usability of our approach in continuous learning scenarios.

2 Background

2.1 Adapters

Adapters (Rebuffi et al., 2018) are light-weight bottleneck layers inserted into a MMT to fine-tune the model for a new task (Houlsby et al., 2019), language (Pfeiffer et al., 2020b) or domain (Bapna and Firat, 2019). The pre-trained weights of the transformer remain fixed and only adapter parameters are updated. This setup prevents catastrophic forgetting (McCloskey and Cohen, 1989) by encapsulating specialized knowledge.

Formally, an adapter module $A_i$ at layer $i$ consists of a down-projection $D_i \in \mathbb{R}^{h \times b}$ of the in-
conditioned on the task and language embeddings, the hypernetwork generates composite adapter layers for the corresponding task-language combination (e.g. NER in Turkish), thereby enabling transfer to arbitrary task-language pairs at test time. Figure 2 provides an overview of our model.

By jointly learning from task and language information, Hyper-X overcomes some of the limitations of prior work: Unlike adapter-based approaches (Pfeiffer et al., 2020b; Üstün et al., 2020) that transfer cross-lingual information only to the task of the task adapter, our model is capable of leveraging supervision—and positive transfer—from both multiple tasks and languages. Moreover, unlike Ponti et al. (2021) who require annotated data in one of the target tasks for each language, Hyper-X is able to perform zero-shot transfer even when there is no annotated data from any of the target tasks, by using MLM as an auxiliary task for each language.

3 Hyper-X

We propose, Hyper-X, an efficient adaptation of a MMT by exploiting multiple sources of information for transfer to an unseen language or task-language pairs. Specifically, Hyper-X learns to combine task and language-specific knowledge in the form of embeddings using a hypernetwork. Conditioned on the task and language embeddings, the hypernetwork generates composite adapter layers for the corresponding task-language combination (e.g. NER in Turkish), thereby enabling transfer to arbitrary task-language pairs at test time. Figure 2 provides an overview of our model.

By jointly learning from task and language information, Hyper-X overcomes some of the limitations of prior work: Unlike adapter-based approaches (Pfeiffer et al., 2020b; Üstün et al., 2020) that transfer cross-lingual information only to the task of the task adapter, our model is capable of leveraging supervision—and positive transfer—from both multiple tasks and languages. Moreover, unlike Ponti et al. (2021) who require annotated data in one of the target tasks for each language, Hyper-X is able to perform zero-shot transfer even when there is no annotated data from any of the target tasks, by using MLM as an auxiliary task for each language.

3.1 A Hypernetwork for Task-Language Adapters

We use a standard hypernetwork as the parameter generator function. However, instead of generating the full model parameters, our hypernetwork generates the parameters for each adapter layer. Concretely, the hypernetwork $\mathcal{H}$ generates adapter parameters where each adapter layer $A_i$ consists of down and up-projection matrices $(D_i, U_i)$:

$$D_i, U_i \triangleq \mathcal{H}(s_i^{(h)})$$

Decoupling Tasks and Languages In Hyper-X, we condition the parameter generation on the input task and language. Therefore, given a combination of task $t \in \{t_1, ..., t_m\}$ and language $l \in \{l_1, ..., l_n\}$, the source embedding contains

![Figure 2: Overview of Hyper-X. The hypernetwork (1) takes the concatenation of task, language and layer embeddings as input and generates a flat parameter vector. Before the final transformation, the source projector network projects the combination of these embeddings to a smaller dimension. The parameter vector is then reshaped and cast to weights of the adapter (2), which are inserted into a transformer layer (3).](image)

When using a hypernetwork, the main model learns weights for a larger main network (Ha et al., 2016). A hypernetwork is a network that generates the where this feed-forward network is followed by a residual link connecting to the input $z_i$.

2.2 Hypernetworks

A hypernetwork is a network that generates the weights for a larger main network (Ha et al., 2016). When using a hypernetwork, the main model learns the desired objective (e.g. classification) whereas the hypernetwork takes an auxiliary input (usually an embedding) that represents the structure of the weights and generates parameters of the main model. A hypernetwork thus enables learning a single parameter space shared across multiple transfer dimensions such as tasks (Mahabadi et al., 2021b) or languages (Platanios et al., 2018) while also allowing input-specific reparametrization.

More concretely, a hypernetwork is a generator function $\mathcal{H}$ that takes an embedding $s_i^{(h)} \in \mathbb{R}^{d_i}$ representing the input sources, and generates the model parameters $\Theta$:

$$\Theta \triangleq \mathcal{H}(s_i^{(h)})$$

While $\mathcal{H}$ can be any differentiable function, it is commonly parameterized as a simple linear transform $(W_h)$ that generates a flat vector with the dimension of $d_h$, which corresponds to the total number of model parameters. $W_h$ is shared across all input sources, enabling maximum sharing.
knowledge from both sources: \( s(h) \approx (t, l) \). We parameterize each task and language via separate embeddings, which enables adaptation to any task-language combination. Task and language embeddings \( (s(t), s(l)) \) are low-dimensional vectors that are learned together with the parameters of the hypernetwork. During training, for each mini-batch we update these embeddings according to the task and language that the mini-batch is sampled from.

**MLM as Auxiliary Task** Hyper-X learns separate tasks and languages embeddings—as long as the task and language have been seen during training. As annotated data in many under-represented languages is limited, we employ MLM as an auxiliary task during training to enable computing embeddings for every language. Moreover, MLM enables a better zero-shot performance for languages that are not included in MMT pre-training (see § 6.2 for a detailed analysis of the impact of MLM).

**Sharing Across Layers** In addition to the task and language embeddings, we learn a layer embedding \( s(i) \) \cite{Mahabadi2021AnS,Ansell2021HyperX} corresponding to the transformer layer index \( i \) where the respective adapter module is plugged in. Since Hyper-X generates an adapter for each Transformer layer, learning independent layer embeddings allows for information sharing across those layers. Moreover, as layer embeddings allow the use of a single hypernetwork for all Transformer layers, they reduce the trainable parameters, i.e., size of the hypernetwork, by a factor corresponding to the number of layers of the main model.

**Combining Multiple Sources** To combine language, task and layer embeddings, we use a simple source projector network \( P_s \) as part of our hypernetwork. This module consisting of two feed-forward layers with a ReLU activation takes the concatenation of the three embeddings and learns a combined embedding \( s(p) \in \mathbb{R}^{d_p} \) with a potentially smaller dimension:

\[
\begin{align*}
    s^{(h)} &= s^{(t)} \oplus s^{(l)} \oplus s^{(i)} \quad (4) \\
    s^{(p)} &= P_s(s^{(h)}) \quad (5)
\end{align*}
\]

where \( s^{(h)} \in \mathbb{R}^{d_h} \) refers to the concatenated embedding before the \( P_s \), with \( d_s = d_t + d_l + d_i \). This component enables learning how to combine source embeddings while also reducing the total number of trainable parameters.

### 4 Experiments

**Dataset and Languages** We conduct experiments on two downstream tasks: part-of-speech (POS) tagging and named entity recognition (NER). For POS tagging, we use the Universal Dependencies (UD) 2.7 dataset \cite{Zeman2020UniversalD} and for NER, we use WikiANN \cite{Pan2017MethodologiesIF} with the train, dev and test splits from \cite{Rahimi2019TheCN}. In addition to these two tasks, we also use masked language modelling (MLM) on Wikipedia articles as an auxiliary task. We limit the number of sentences from Wikipedia to 100K for each language, in order to control the impact of dataset size and to reduce the training time.

For the language selection, we consider: (i) typological diversity based on language family, script and morphosyntactic attributes; (ii) a combination of high-resource and low-resource languages based on available data in downstream task; (iii) presence in the pre-training data of mBERT; and (iv) presence of a language in the two task-specific datasets.\(^2\) We provide the details of the language and dataset selection in Appendix A.

**Experimental Setup** We evaluate Hyper-X for zero-shot transfer in three different settings: (1) **English single-task**, where we train the models only on English data for each downstream task separately. (2) **English multi-task**, where the models are trained on English POS and NER data at the same time. (3) **Mixed-language multi-task**, where we train the models in a multi-task setup, but instead of using only English data for both POS and NER, we use a mixture of task-language combinations. In order to measure zero-shot performance in this setup, following \cite{Ponti2021AEO} we create two different partitions from all possible language-task combinations in such a way that a task-language pair is always unseen for one of the partitions (e.g. NER-Turkish and POS-Arabic in Figure 1). Details of partitions and our partitioning strategy are given in Appendix A.

### 4.1 Baselines and Model Variants

**mBERT** \cite{Devlin2019BERTPO} is a MMT that is pre-trained for 104 languages. We use mBERT by fine-tuning all the model parameters on the
available sources. As this standard approach enables cross-lingual transfer from both a single source or a set of language-task combinations, we compare it to Hyper-X in all three settings. Moreover, we use mBERT as the base model for both Hyper-X and the other baselines.

**MAD-X** (Pfeiffer et al., 2020b) is an adapter-based modular framework for cross-lingual transfer learning based on MMTs. It combines a task-specific adapter with language-specific adapters that are independently trained for each language using MLM. We train MAD-X language adapters on the same Wikipedia data that is used for Hyper-X, for all languages with a default architecture. Finally, for the mixed-language setup, as the original MAD-X does not allow standard multi-task training, we train the task adapters by using multiple source languages but for NER and POS separately. We call this model MAD-X MS.

**Parameter Space Factorization** (Ponti et al., 2021) is a Bayesian framework that learns a parameter generator from multiple tasks and languages for the softmax layer on top of a MMT. However, if a language lacks annotated training data, this model cannot learn the required latent variable for the corresponding language. Therefore, we evaluate this baseline only for the mixed-language multi-task setting using the same partitions as Hyper-X. We use the original implementation with default hyper-parameters and low-rank factorization.

**Model Variants** We evaluated two variants of Hyper-X in order to see the impact of Hypernetwork size: Hyper-X Base model fine-tunes 76m parameters ($d_s = 192$), compatible with MAD-X in terms of total number of trainable parameters, and Hyper-X Small updates only 13m parameters ($d_s = 32$). Table 3 shows the parameter counts together with the corresponding runtime.

### 4.2 Training Details

For all the experiments, we used a batch size of 32 and a maximum sequence length of 256. We trained Hyper-X for 100,000 updates steps by using a linearly decreasing learning rate of 1e-4 with 4000 warm-up steps. We evaluated checkpoints every 5000 steps, and used the best checkpoint w.r.t. the average validation score for testing. As for baselines, we trained mBERT and MAD-X tasks adapters for 20 epochs by using learning rate of 1e-5 and 1e-4 respectively with the same scheduler and warm-up steps. Since MAD-X requires prerequisite language adapters, we trained language adapters for 100,000 steps for each language separately.

In terms of model size, we use a bottleneck dimension of 256 to learn adapters for Hyper-X. Similarly, we train language and adapters with dimension of 256 and 48 for MAD-X to create a comparable baseline. In Hyper-X, as input to the hypernetwork, dimensions for task, language and layer embeddings are all set to 64 (total 192). During training, we create homogeneous mini-batches for each task-language pair to learn the corresponding embeddings together with the hypernetwork. Moreover, following Mahabadi et al. (2021b), we also update the original layer-norm parameters. During multi-task training, we use temperature-based sampling with $T = 5$ to balance each task-language pair during training (See Appendix § B.1 for details).

### 5 Zero-shot Transfer Results

Table 2 shows the aggregate zero-shot results in NER and POS tagging respectively. In addition to the average scores across all 15 zero-shot languages, we show the average of the 8 ‘seen’ and 7 ‘unseen’ languages separately with respect to language coverage of mBERT. We present results for English single-task, English multi-task and Mixed-language multi-task settings.

Overall, Hyper-X Base performs on par with the strongest baseline when transferring from English. In the presence of additional sources, such as a mixture of task-language pairs, Hyper-X outperforms both mBERT and parameter space factorization (PSF). In comparison to MAD-X, Hyper-X generally performs better on seen languages. We relate this to the unified hypernetwork enabling maximum sharing between languages and higher utilization of the pre-trained capacity in contrast to the isolated adapters. On unseen languages, Hyper-X is outperformed by MAD-X in most cases. However, we emphasize that MAD-X requires training separate language adapters for each new language,
| Source    | Method   | Time          | #Params | Named-Entity Recognition | Part-of-Speech Tagging |
|-----------|----------|---------------|---------|-------------------------|------------------------|
|           |          |               | SEEN   | UNSEEEN | ALL   | SEEN   | UNSEEEN | ALL   |
| English   | mBERT    | 177m / 2h     | 53.8   | 40.4    | 47.6  | 65.8   | 47.7    | 57.3  |
|           | MAD-X    | 76m / 116h    | 54.3   | 51.1    | 52.8  | 67.7   | 62.6    | 65.4  |
| (Single-Task) | Hyper-X Small | 13m / 16h   | 54.2   | 47.7    | 51.2  | 66.5   | 57.9    | 62.5  |
|           | Hyper-X Base | 76m / 18h | 54.4   | 50.7    | 52.7  | 67.8   | 58.7    | 63.5  |
| English   | mBERT    | 177m / 2h     | 56.4   | 48.7    | 52.8  | 67.2   | 54.7    | 61.4  |
| (Multi-Task) | Hyper-X Small | 13m / 16h   | 52.2   | 49.3    | 50.8  | 65.1   | 57.9    | 61.7  |
|           | Hyper-X Base | 76m / 18h | 54.4   | 51.1    | 52.9  | 67.0   | 59.7    | 63.6  |
|           | mBERT    | 185m / 4h     | 58.1   | 54.1    | 56.2  | 70.4   | 53.8    | 62.7  |
|           | PSF      | 185m / 4h     | 58.1   | 54.1    | 56.2  | 70.4   | 53.8    | 62.7  |
|           | MAD-X MS | 76m / 116h    | 62.4   | 62.2    | 62.3  | 70.7   | 67.0    | 69.0  |
| Mixed-Language | Hyper-X Small | 13m / 16h   | 62.0   | 58.3    | 60.3  | 70.7   | 63.2    | 67.2  |
| (Multi-Task) | Hyper-X Base | 76m / 18h | 63.3   | 61.0    | 62.3  | 71.5   | 63.8    | 67.9  |

Table 2: Zero-shot cross-lingual transfer results averaged over 3 runs on Named-Entity Recognition (NER; F1) and Part-of-Speech Tagging (POS; Accuracy) for mBERT, MAD-X (Pfeiffer et al., 2020b), parameter space factorization (PSF; Ponti et al., 2021) and Hyper-X. We highlight the best results per-setting in bold. We also report the total number of parameters and fine-tuning time for all models. Note that Hyper-X corresponds to a single model trained for each partition while MAD-X consists of *N* independently trained adapters for each task and language. MAD-X MS refers to an adapted version of the original model trained on multiple source languages but each task separately.

which makes it considerably less resource-efficient than Hyper-X (see § 6.1).

**English Single-Task** When English is used as the only source language for each task separately, Hyper-X (Base) performs on par with MAD-X for NER (52.7 vs 52.8 F1) but falls behind for POS tagging (63.5 vs 65.4 Acc.) on average. Both models significantly outperform mBERT. Looking at the individual language results, Hyper-X performs slightly better on ‘seen’ languages compared to MAD-X in NER and POS tagging respectively. For ‘unseen’ languages, both MAD-X and Hyper-X benefit from MLM, which results in large improvements with respect to mBERT. Between the two models, MAD-X achieves a higher average score in both NER and POS tagging.

**English Multi-Task** In a multi-task setting where only English data is available, fine-tuning mBERT for both target tasks at the same time gives mixed results compared to single-task training—in line with previous findings noting catastrophic forgetting and interference in MMTs (Wang et al., 2020). Hyper-X Base, on the other hand, shows a small but consistent improvement on the majority of languages, with 0.2 (F1) and 0.1 (Acc.) average increase in NER and POS tagging respectively. This confirms that Hyper-X is able to mitigate interference while allowing for sharing between tasks when enough capacity is provided.4

**Mixed-Language Multi-Task** In this setting, a mixture of language data is provided for NER and POS via two separate training partitions while keeping each task-language pair unseen in one of these partitions. All the models including mBERT achieve better zero-shot scores compared to the previous settings. Among the baselines, parameter space factorization (PSF) gives a larger improvement compared to mBERT on both tasks, indicating the importance of task- and language-specific parametrization for adapting a MMT. Hyper-X Base produces the largest performance gain among the models that trains only a single model: it achieves 9.0 (F1) and 4.3 (Acc.) average increase for NER and POS. Although both PSF and Hyper-X enable adaptation conditioned on a mixture of task and language combinations, we relate the difference between PSF and Hyper-X to the contrast in parameter generation. PSF only generates parameters of the softmax layer and is thus unable to adapt deeper layers of the model. Hyper-X, on the other hand, generates adapter layer parameters inserted throughout the model, which provide a higher degree of adaptation flexibility. Hyper-X outperforms PSF particularly on unseen languages as it benefits from MLM as an auxiliary task.

4MAD-X learns independent adapters for each target task, which does not allow for positive cross-task transfer.
Table 3: Compute efficiency with respect to number of fine-tuned parameters and training time for mBERT, PSF, MAD-X and Hyper-X. Training time includes both NER and POS-tagging. For MAD-X, the total number of parameters and training time is calculated for 16 (l) languages and 2 (t) tasks.

| Model      | #Params. | Training Time |
|------------|----------|---------------|
| mBERT      | 177m     | 2h            |
| PSF        | 185m     | 4h            |
| MAD-X      | 76m      | 116h          |
| Language Adapters | 4.7m x l | 7h x l |
| Task Adapters | 0.9m x t | 2h x t |
| Hyper-X Small | 13m     | 16h           |
| Hyper-X Base   | 76m      | 18h           |

Finally, Hyper-X tends to perform slightly better on seen languages compared to the adapted multi-source version of MAD-X. However, MAD-X outperforms Hyper-X on unseen languages by 1.2 (F1) and 2.8 (Acc.) for NER and POS respectively. Besides the expected benefits of independently trained language adapters in MAD-X, we relate this to the limited cross-task supervision for unseen languages in Hyper-X for this setting. Especially, when the target task is POS, most of the unseen languages have only 100 sentences available in NER dataset, which leaves only a little margin for improvements.

6 Analysis

6.1 Parameter and Time Efficiency

Table 3 shows the fine-tuned parameter counts and the training time required for the baselines and Hyper-X models. Unlike mBERT, PSF and Hyper-X, MAD-X consists of 16 and 2 independently trained language and task adapters respectively. In terms of parameter efficiency, MAD-X and Hyper-X Base models correspond to 43% of mBERT’s parameters. However, in terms of training time, Hyper-X Base is trained only once for about 18 hours, as opposed to MAD-X’s considerably high total training time (116 hours in total). Thus, considering the competitive zero-shot performances across different languages and settings, Hyper-X Base provides a better efficiency-performance trade-off. Furthermore, in the case of adding more languages, MAD-X’s parameter count and training time increase linearly with the number of new languages, while Hyper-X’s computational cost remains the same.

As Hyper-X model variants, we evaluated two different sizes of the source embedding (d_s; 32→192). Although Hyper-X Small is much more parameter-efficient (7.2% of mBERT’s parameters) and takes slightly less time to train (16h), its zero-shot performance is significantly lower than the base model, especially for unseen languages. Nevertheless, Hyper-X Small remains a valid alternative for particularly ‘seen’ languages.

6.2 Impact of Auxiliary MLM Training

Figure 3 demonstrates the impact of auxiliary MLM training in Hyper-X Base for the mixed-language multi-task setting. As this setting provides training instances for each task and language, we evaluated the impact of MLM by removing the corresponding Wikipedia data first for ‘seen’ languages, then for ‘all’ languages. As shown in the figure, although the availability of MLM data slightly increases seen language performance, it mainly boosts the scores in unseen languages: +6.2 F1 and +10.5 Acc. for NER and POS respectively. Furthermore, when MLM data is removed for only seen languages, Hyper-X can mostly recover performance on seen languages, confirming the dominant effect of MLM on unseen languages.

6.3 Impact of Source Languages

In the mixed-language multi-task setting, we deliberately avoid grouping languages from same families to different partitions, in order to restrict the transfer from the same-language family instances, and to observe the effect of cross-task supervision. However, we also evaluate the impact of source languages in this setup, to measure the degree of potential positive transfer. To this end, we switched the partitions of kk, mt, yue, so that all of them will likely benefit from a high-resource language.
from the same family for the same target task. Figure 4 and 5 shows the aggregated results in both Hyper-X Base and mBERT. Firstly, both models benefit from positive transfer. Secondly, although the relative increase in mBERT is slightly higher Hyper-X still outperforms mBERT with a large margin, showing the robustness of our model with regard to different partitions.

6.4 Few-shot Transfer

Fine-tuning an MMT with a few target instances has been shown to increase zero-shot performances (Lauscher et al., 2020b). Therefore, we evaluate Hyper-X for few-shot transfer on 5 languages—3 of which are high-resource and covered by mBERT and 2 are low-resource and unseen. To this end, we further fine-tune Hyper-X and the corresponding baselines that are trained initially in the English multi-task by using 5, 10, 20, and 50 training instances for each language separately on NER and POS-tagging (see details in Appendix §D).

Figure 6 presents the average results comparing mBERT to MAD-X. Similar to the zero-shot results, on seen languages, Hyper-X constantly provides better adaptation than both baselines for NER and POS. On unseen languages, MAD-X gives the best result on average. This is because MAD-X starts with better initial representations for Maltese and Uyghur. When more samples are provided Hyper-X reduces the initial gap. Overall, Hyper-X consistently achieves the best or competitive performance on the majority of the experiments, except ‘unseen’ languages for POS tagging, showing the effectiveness of our approach beyond the standard zero-shot transfer. Taken together with the parameter and training efficiency, these results show that Hyper-X can be easily extended to new languages without incurring large computing costs.

7 Related Work

Adapters As a parameter-efficient alternative to standard fine-tuning, adapters have been used for quick training (Rücklé et al., 2021), multi-task learning (Stickland and Murray, 2019) and knowledge composition (Pfeiffer et al., 2021a; Wang et al., 2021; Poth et al., 2021). Moreover, Mahabadi et al. (2021a) and He et al. (2022a) extended adapters for better performance with fewer parameters. In the context of multilingual transfer, adapters enable allocation of additional language-specific capacity, thereby mitigating the ‘curse of multilinguality’ (Üstün et al., 2020). Such language adapters (Pfeiffer et al., 2020b; Ansell et al., 2021) achieve high zero-shot results when combined with task adapters and enable generalization to languages unseen during pre-training via MLM-based adaptation (Pfeiffer et al., 2021b). Philip et al. (2020) and Üstün et al. (2021) also used monolingual adapters for zero-shot and unsupervised NMT.

Hyernetworks in NLP Tay et al. (2021) propose a multi-task model that uses a hypernetwork to condition on input to learn task-specific reparametrizations. Similarly, Mahabadi et al. (2021b) generate task-specific adapters via a hypernetwork. Recently, He et al. (2022b) use a hypernetwork to generate prompts. For multilingual learning, where the input sources correspond to language embeddings, Üstün et al. (2020) and Ansell et al. (2021) learn these embeddings from the typological feature vectors of languages, enabling generalization to unseen languages based on a hypernetwork. In a similar spirit to our work, parameter space factorization (PSF; Ponti et al., 2021), learns task and language-specific embeddings from seen task-language combinations. However, unlike our model, these embeddings are used
We have proposed Hyper-X, a novel approach for multi-task multilingual transfer learning, based on a unified hypernetwork that leverages heterogeneous sources of information, such as multiple tasks and languages. By learning to generate composite adapters for each task-language combinations that modify the parameters of a pre-trained multilingual transformer, Hyper-X allows for maximum information sharing and enables zero-shot prediction for arbitrary task-language pairs at test time. Through a number of experiments, we demonstrate that Hyper-X is competitive with the state-of-the-art when transferring from a source language. When a mixture of tasks and languages is available, Hyper-X outperforms several strong baselines on many languages, while being more parameter and time efficient. Finally, we show that for few-shot transfer, Hyper-X is a strong option with a less computing cost than baselines for the initial task adaptation.

9 Limitations

Firstly, although our experiments show the potential of Hyper-X to benefit from multiple tasks for zero-shot transfer, so far we evaluated our model on a limited set of tasks: NER and POS-tagging, which may limit the generalizability of our model to other tasks.

Secondly, for the few-shot transfer, we limit our experiments to languages that we learn via MLM and to existing tasks. Our work does not include languages without MLM data as well as completely new tasks. Learning the task and language embeddings separately, however, creates a possibility to interpolate existing embeddings for new languages or new tasks, which especially may work for the few-shot learning. We leave exploration of these two limitations to future work.

Acknowledgements

We would like to thank Noah Constant, Asa Cooper Stickland and the anonymous reviewers for their helpful feedback on a previous version of this paper. We also would like to thank the Center for Information Technology of the University of Groningen for providing access to the Peregrine HPC cluster.

References

Alin Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. MAD-G: Multilingual adapter generation for efficient cross-lingual transfer. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4762–4781, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8440–8451.

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

David Ha, Andrew Dai, and Quoc V Le. 2016. Hypernetworks. In International Conference on Learning Representations.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022a. Towards a unified view of parameter-efficient transfer learning. In Proceedings of ICLR 2022.

Yun He, Huaxi Yu, Steven Zheng, Yi Tay, Jai Gupta, Yu Du, Vamsi Aribandi, Zhe Zhao, YaGuang Li, Zhao Chen, Donald Metzler, Heng-Tze Cheng, and Ed H. Chi. 2022b. HyperPrompt: Prompt-based Task-Conditioning of Transformers. arXiv preprint arXiv:2203.00759.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In International Conference on Machine Learning, pages 2790–2799.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization. In Proceedings of ICLR 2020.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020a. From Zero to Hero: On the Limitations of Zero-Shot Cross-Lingual Transfer with Multilingual Transformers. In Proceedings of EMNLP 2020.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020b. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4483–4499, Online. Association for Computational Linguistics.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021a. Compacter: Efficient low-rank hypercomplex adapter layers. In Advances in neural information processing systems.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021b. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. 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), pages 565–576, Online. Association for Computational Linguistics.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In Psychology of learning and motivation, volume 24, pages 109–165. Elsevier.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atuxa, Liesbeth Augustinus, Elana Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gossie Bouma, Sam Bowman, Adrian Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gilsen Cebiroğlu Eryiğit, Flavio Massiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čelpló, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, MiriamConnor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantzaz Díaz de Ibarra, Carl Dickerson, Peter Dirix, Kaja Dobroviljic, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richard Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdóšová, Daniel Galbraith, Marcos Garcia, Mo Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Menduh Gökçärkem, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Matias Grioni, Normunds Grūzits, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajic, Jan Hajíc jr., Lính Hà Mây, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jarošava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Huang, Radu Ion, Elena Irmita, Olájídè Ishola, Tomáš Jelinek, Anders Johannsen, Fredrik Jørgensen, Hüner Kalkşarka, Sylvin Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Kamil Kopacewicz, Natalia Kotsyba, Simon Krek, Sookyoun Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê Hông, Alessandro Lanci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, Kyung-Tae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashesvskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Măranduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, Andrés Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălina Mititelu, Yusuke Miyao, Simonetta Montemagni, Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Măranduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, Andrés Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălina Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Shinshuke Mori, Bjartur Mortensen, Bohdán Moskal'evskyi, Kadri Muischnek, Yugo Murawaki, Kaili Mürissep, Pinky Nainwani, Juan Ignacio Navarro Horriñacek, Anna Nedoluzhko, Gunta
Nešpore-Bērzaknife, Luong Nguyên Thi, Huyễn Nguyên Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Sínta Qija, Adédayo Olúkkun, Mai Omura, Petia Osenova, Robert Östling, Lilija Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guylherme Paulino-Passos, Siyao Peng, Cen-C mango Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pittler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalinia, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolalainen, Sampo Pyysalo, Andrea Räbbis, Alexandre Rade-maker, Loghanathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mhykailo Romannenko, Rudolf Rosa, Davide Rovati, Valentin Rößler, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulite, Yanin Sawanakunanan, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Moğan Seraj, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Simonková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Ureová, Larraitz Uria, Hans Uszko-reit, Sowmya Vajjala, Daniel van Niekerk, Gert-Jan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Jing Xian Wang, Jonathan North Washington, Seiy Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 2018. Universal dependencies 2.3. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (UFAL), Faculty of Mathematics and Physics, Charles University.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021a. AdapterFusion: Non-destructive task composition for transfer learning. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterhub: A framework for adapting transformers. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations, pages 46–54, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. 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.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021b. UNKs everywhere: Adapting multilingual language models to new scripts. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4465–4470, Online. Association for Computational Linguistics.

Emmanuel Antonios Platanios, Minmaya Sachan, Graham Neubig, and Tom Mitchell. 2018. Contextual parameter generation for universal neural machine translation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 425–435.

Edoardo M. Ponti, Ivan Vulić, Ryan Cotterell, Marinella Parovic, Roi Reichart, and Anna Korhonen. 2021. Parameter space factorization for zero-shot learning across tasks and languages. Transactions of the Association for Computational Linguistics, 9:410–428.

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to pre-train on? Efficient intermediate task selection. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 10585–10605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Afsin 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, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2018. Efficient parametrization of multi-domain deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8119–8127.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. AdapterDrop: On the efficiency
of adapters in transformers. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7930–7946, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sebastian Ruder, Matthew E Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials, pages 15–18.

Asa Cooper Stickland and Iain Murray. 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. In International Conference on Machine Learning, pages 5986–5995.

Stephanie Strassel and Jennifer Tracey. 2016. Lorelei language packs: Data, tools, and resources for technology development in low resource languages. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16), pages 3273–3280.

Yi Tay, Zhe Zhao, Dara Bahri, Donald Metzler, and Da-Cheng Juan. 2021. HyperGrid Transformers: Towards A Single Model for Multiple Tasks. In Proceedings of ICLR 2021.

Ahmet Üstün, Alexandre Berard, Laurent Besacier, and Matthias Gallié. 2021. Multilingual unsupervised neural machine translation with denoising adapters. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6650–6662, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDAdapter: Language adaptation for truly Universal Dependency parsing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2302–2315, Online. Association for Computational Linguistics.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1405–1418, Online. Association for Computational Linguistics.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4438–4450, Online. Association for Computational Linguistics.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, Xiaohong Li, Zhi Yuan Lim, Sidik Soleman, Rahmad Mahendra, Pascale Fung, Syafri Bahar, and Ayu Purwarianti. 2020. IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding. In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 843–857, Suzhou, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45. Online. Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kenneth Ajede, Gabriele Aleksandričič, Ika Alfína, Lena Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Hórunn Arnardóttir, Gashaw Artjue, Naira Araiswari Arwidarasti, Masayuki Asahara, Luma Ateyah, Furkan Atmaca, Mohammed Attila, Aitzibar Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verónica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adrieane Boyd, Kristina Brokátí, Aljoscha Burchardt, Marie Candido, Bernard Caron, Gauthier Caron, Tatiana Cavalcanti, Gülşen Çebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Cépló, Sávás Cetin, Özlem Çetinoğlu, Fabrizio Chalub, Ethan Chi, Yongseok Cho, Jinho Choi, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45. Online. Association for Computational Linguistics.
A Language Selection

Table 4 shows that the details for languages such as language code, UD treebank id and language family. For POS tagging, we use the Universal Dependencies (UD) 2.7 dataset (Zeman et al., 2020) and for NER, we use WikiANN (Pan et al., 2017) with the train, dev and test splits from Rahimi et al. (2019). To partition languages for the mixed-language multi-task setting, we group languages from the same families into the same partitions to avoid a strong supervision from the same language family when evaluating zero-shot predictions for unseen task-language combinations. When there is no available training data in the target treebank, we use the test split for the mixed-language multi-task setting.

B Experimental Details

B.1 Impact of Sampling

Hyper-X is a single model that is trained at once for multiple languages and task simultaneously. However, as the amount of total MLM training data is considerably larger than NER and POS-tagging data, we experimented with two different sampling methods: size proportional sampling and temperature-based sampling ($t = 5$). For the temperature-based sampling, we independently sample a batch for each task-language combination. Figure 7 shows the impact of different sampling methods on the zero-shot performance for ‘seen’, ‘unseen’ language groups together with average over all languages. As seen, temperature-based sampling greatly increase performance for all language groups on both NER and POS-tagging. This suggest that when MLM data does not restricted by sampling, it highly influences the learning objective which results a catastrophic forgetting on the target tasks.

B.2 Implementation and Computing Infrastructure

All the experiments are conducted using Tesla V100 GPUs. We did not use parallel training on multiple GPUs, so each experiment was conducted on a single GPU. Parameters that are fine-tuned for each model and total runtime are reported in the section (§ 6.1). We implemented Hyper-X by using Transformers library (Wolf et al., 2020) and the code will be released upon publication. We used adapterhub (Pfeiffer et al., 2020a) for MAD-X, and

Figure 7: Impact of sampling for SEEN, UNSEEN language groups on NER and POS tagging.

the original repository for parameter space factorization (Ponti et al., 2021). Hyper-parameters that are used in experiments are given in the section 4. We did not conduct a hyper-parameter search due to the computational limitations, and used the reference values in most cases: only the dimension for language adapters in MAD-X is changed to match with the same parameter count of Hyper-X. Finally for mBERT, we did a preliminary experiments with learning rate of 1e-4 and 1e-5, and pick the latter one as it produced better performance.

C Detailed Results

The results that are averaged over 3 runs for each language are given in Table 6

D Few Shot Experiments

For the few-shot transfer experiments, we fine-tune each model for 50 epochs with the same hyper-parameters. We disable the learning rate decay as only a few training instances are provided to the models. Note that, in these experiments, we always start with the models that are already trained in the zero-shot setting and perform fine-tuning for each language and task separately. For the selection of training samples, we randomly sample instances regardless of the labels, as the initial models are already trained for these tasks on English data. Table 5 show that few-shot results for NER and POS-tagging respectively.
### Table 4: Languages that are used in the experiments, together with corresponding language code, UD treebank and language families. We used WikiANN (Pan et al., 2017; Rahimi et al., 2019) and UD version 2.7 (Zeman et al., 2020) for NER and POS-tagging respectively.

| Language  | Code | UD Treebank | Family               | Train | Dev | Test | Train | Dev | Test |
|-----------|------|--------------|----------------------|-------|-----|------|-------|-----|------|
| English   | en   | EWT          | IE, Germanic         | 20000 | 1000 | 10000| 12543 | 2002| 2077 |
| Arabic    | ar   | PADT         | Afro-Asiatic, Semitic| 20000 | 1000 | 10000| 6075  | 909 | 680  |
| Breton    | br   | KEB          | IE, Celtic           | 1000  | 1000| 10000 | 3997  | 500 | 500  |
| Chinese   | zh   | GSD          | Sino-Tibetan         | 20000 | 10000| 10000| 3664  | 988 | 983  |
| Icelandic | is   | PUD          | IE, Germanic         | 1000  | 1000| 10000 | -     | -   | -    |
| Kazakh    | kk   | KTB          | Turkic, Northwestern | 1000  | 1000| 10000 | 32    | -   | 1047 |
| Tamil     | ta   | TTB          | Dravidian, Southern  | 15000 | 10000| 10000| 400   | 80  | 120  |
| Turkish   | tr   | IMST         | Turkic, Southwestern | 20000 | 10000| 10000| 3664  | 988 | 983  |
| Yoruba    | yo   | YTB          | Niger-Congo, Defoid  | 100   | 1000| 10000 | -     | 318 |     |
| Faroese   | fo   | OFT          | IE, Germanic         | 100   | 1000| 10000 | -     | -   | 1208 |
| Guarani   | gn   | Thomas       | Tupian, Tapi-Guarani | 100   | 1000| 10000 | -     | -   | 98   |
| Upper Sorbian | yo | UFAL | IE, Slavic          | 100   | 1000| 10000 | -     | -   | 23   |
| Maltese   | mt   | MUDT         | Afro-Asiatic, Semitic| 100   | 1000| 10000| 1123  | 433 | 518  |
| Sanskrit  | sa   | UFAL         | Indic                | 100   | 1000| 10000 | -     | -   | 230  |
| Uyghur    | ug   | UDT          | Turkic, Southeastern | 100   | 1000| 10000| 1656  | 900 | 900  |
| Cantonese | yue  | HK           | Sino-Tibetan         | 20000 | 10000| 10000| -     | -   | 1004 |

### Table 5: Per language results for few-shot experiments, where models are further fine-tuned with a few training instances (0, 5, 10, 20, 50) from NER and POS datasets. For the language selection, ar, tr, zh are covered by mBERT and mt, ug are unseen.

| Language  | mBERT | MAD-X | Hyper-X Small | Hyper-X Base |
|-----------|-------|-------|---------------|--------------|
| ar, tr, zh|       |       |               |              |
| mt, ug    |       |       |               |              |

| Language  | mBERT | MAD-X | Hyper-X Small | Hyper-X Base |
|-----------|-------|-------|---------------|--------------|
| ar, tr, zh|       |       |               |              |
| mt, ug    |       |       |               |              |

Table 5: Per language results for few-shot experiments, where models are further fine-tuned with a few training instances (0, 5, 10, 20, 50) from NER and POS datasets. For the language selection, \textit{ar, tr, zh} are covered by mBERT and \textit{mt, ug} are unseen.
Table 6: Zero-shot cross-lingual transfer results averaged over 3 runs for Named-Entity Recognition (NER; F1) and Part-of-Speech Tagging (POS; Accuracy) for mBERT (mB), MAD-X (MX) and parameter space factorization (PSF) models, together with Hyper-X Small (HX.32) and Base (HX.192). Superscripts denote the partitioning that is used for mixed-language multi-task setting.

| Language | NER Single-Task | NER Multi-Task | POS Single-Task | POS Multi-Task |
|----------|----------------|----------------|----------------|----------------|
| mb       | 84.2 81.6 83.6 | 83.8           | 82.6           | 82.6           |
| mx       | 81.8 82.2 83.8 | 83.7           | 79.2           | 79.2           |
| hx.32    | 83.8 83.7      |                |                |                |
| mx       | 85.5 93.4 53.5 | 47.8           | 49.2           |                |
| hx.32    | 84.2 80.0 54.7 | 50.0           |                |                |
| mb       |                |                |                |                |
| psf      | 70.9 72.3 74.7 | 76.1           |                |                |
| mx       | 55.4 57.1 59.2 | 56.4           |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
|               | 70.6 73.5 77.5 | 77.3           | 80.2           |                |
|          |                |                |                |                |
| mb       |                |                |                |                |
| psf      | 57.3 52.2 60.4 | 61.1           |                |                |
| mx       | 77.2 78.2 78.9 | 80.3           | 82.7           |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |
| mb       |                |                |                |                |
| psf      |                |                |                |                |
| mx       |                |                |                |                |
| hx.32    |                |                |                |                |