

The Hidden Space of Transformer Language Adapters

Jesujoba O. Alabi¹ Marius Mosbach² Matan Eyal³ Dietrich Klakow¹ Mor Geva⁴

¹Saarland University, Saarland Informatic Campus

²Mila, McGill University ³Google Research ⁴Tel Aviv University

jalabi@lsv.uni-saarland.de marius.mosbach@mila.quebec morgeva@tauex.tau.ac.il

Abstract

We analyze the operation of transformer language adapters, which are small modules trained on top of a frozen language model to adapt its predictions to new target languages. We show that adapted predictions mostly evolve in the source language the model was trained on, while the target language becomes pronounced only in the very last layers of the model. Moreover, the adaptation process is gradual and distributed across layers, where it is possible to skip small groups of adapters without decreasing adaptation performance. Last, we show that adapters operate on top of the model’s frozen representation space while largely preserving its structure, rather than on an “isolated” subspace. Our findings provide a deeper view into the adaptation process of language models to new languages, showcasing the constraints imposed on it by the underlying model and introduces practical implications to enhance its efficiency.¹

1 Introduction

The adaptation of pre-trained transformer-based language models (LMs) to new languages has become a widely adopted practice in natural language processing (NLP). A prominent approach for achieving this is to train small feed-forward network neural modules, called *adapters*, on top of the LM layers, while freezing the LM parameters (Houlsby et al., 2019). To successfully adapt an LM to a new language, adapters should introduce changes that steer the prediction but still can be processed by subsequent layers.

Adapters have demonstrated impressive capabilities in zero-shot cross-lingual settings (Pfeiffer et al., 2020b; Lee et al., 2022; Parović et al., 2022), multi-task learning (Pfeiffer et al., 2021a), and integrating knowledge into pre-trained LMs (Lauscher

et al., 2020; Wang et al., 2021). However, despite this vast success, how LM predictions are being adapted internally is largely unknown. Nonetheless, a better understanding of the internal operation of LM adapters would be valuable to inform language selection for multilingual pre-training and for designing more effective adaptation approaches.

In this work, we tackle the question of how language adapters work, while focusing on LM predictions adapted from one or more languages (source) to another language (target). For our study, we train an auto-regressive decoder-only LM from scratch on English and then adapt it separately to four different target languages: German, French, Hebrew and Arabic (§2). To demonstrate the generality of our findings, we additionally experiment with bilingual models as well as mBERT (Devlin et al., 2019), a massively multilingual LM pre-trained on 104 languages.

First, we consider the sequence of hidden representations across the layers as an information stream being evolved through additive updates (Elhage et al., 2021), and analyze the adapters’ updates to the prediction (§3). We find that throughout most of the inference pass, the prediction is evolved in the source language and only in the very last layers the target language abruptly becomes pronounced. Then, we explore the contributions of individual adapters (§4) and observe that adapters introduce gradual updates, which often can be canceled without any decrease in performance, while the last few adapter layers are critical for adaptation success.

Based on these findings, we explore two alternative hypotheses for how adapters interact with the underlying frozen LM (§5). Either adapters operate in an “isolated” subspace of the representation space that is unused for predictions in the source language, or they operate on top of the model’s representation space, while preserving its structure. We test the first hypothesis by training sparse

¹We release our code and models publicly at <https://github.com/uds-lsv/hidden-space-adapters>.

probing classifiers to detect features that change the most during adaptation and then intervene on these features. We observe that while the identified features indeed deteriorate adaptation performance upon intervention, intervening on random features also leads to substantial drops in adaptation performance, and therefore refute the first hypothesis.

To test the second hypothesis we analyze principal components in the representation space of the monolingual base model as well as the adapted models. For different properties such as part-of-speech, number, and tense, we observe a strong alignment between the original representation space and its adapted counterpart.

In summary, we provide novel insights into the prevalent language adaptation process of pre-training LMs, showing that adapted predictions are gradually evolved on top of the representation space of the source language, while being shifted towards the target language only at the end of the computation and that adapters operate on top of the existing structure of the pre-trained model’s representation space. Our work not only provides a first step towards a better understanding of language model adaptation but also opens interesting avenues for future work on making language adaptation more efficient.

2 Experimental setup

We employ a controlled setting of pre-training our own LMs from scratch, and then adapting them to a target language by training language adapters (Pfeiffer et al., 2020b). This is important because some of our experiments require language identification based on sub-tokens. Achieving this with existing multilingual LMs is challenging due to the diverse languages on which the models and their tokenizers have been trained. Where possible, we extend our experiments to mBERT (Devlin et al., 2019), a multilingual LM trained on 104 languages.

Models As our base model, we pre-trained an auto-regressive decoder-only LM on English (en) texts. Our model follows the GPT-NeoX architecture (Black et al., 2022) with $L = 24$ decoder layers, 16 attention heads per layer, and a hidden dimensionality of 1024. We used a vocabulary size of 250K², which is large enough to support multiple languages, and trained a BPE tokenizer (Wang et al., 2019) from scratch

²Resulting in a total of 814M trainable parameters.

on a combination of sentences sampled from these languages. For our experiments in §3, we additionally train a bilingual model on English and a small fraction of German data.

Adaptation to a new language To adapt the model, we trained a separate set of adapters (via language modeling) for each of the following target languages: Arabic (ar), English (en), French (fr), German (de), and Hebrew (he). We choose German and French because of their high similarity to each other, and English and French having a higher lexical overlap than English and German. The choice of Hebrew and Arabic is motivated by the fact that they are relatively similar to each other while highly dissimilar to English, in part by their different non-Latin scripts. Further details on our pre-training and adaptation procedure are provided in Appendix A.

Adapter setup For most of our experiments, we focus on the adapter architecture proposed by Pfeiffer et al. (2020b), which is the most commonly used technique for multilingual LM adaptation and train using its default hyperparameters³. This technique relies on placing an adapter layer in each LM block. Specifically, for every transformer decoder block (Vaswani et al., 2017), an adapter is stacked on top of the feed-forward network (FFN), such that it gets as input the hidden states after the FFN layer and outputs an update to every hidden state that is added via a residual connection. Formally, let t_1, \dots, t_N denote an input sequence consisting of N tokens, the hidden representation $\mathbf{x}_{\text{out}}^{l,i} \in \mathcal{R}^d$ at the i -th position after the l -th block is obtained by:

$$\mathbf{x}_{\text{attn}}^{l,i} = \mathbf{x}_{\text{out}}^{l-1,i} + \text{self-attn}(\mathbf{X}_{\text{out}}^{l-1})_i \quad (1)$$

$$\mathbf{x}_{\text{ffn}}^{l,i} = \mathbf{x}_{\text{attn}}^{l,i} + \text{feed-forward}(\mathbf{x}_{\text{attn}}^{l,i}) \quad (2)$$

$$\mathbf{x}_{\text{out}}^{l,i} = \mathbf{x}_{\text{ffn}}^{l,i} + \text{adapter}(\mathbf{x}_{\text{ffn}}^{l,i}) \quad (3)$$

where $\mathbf{X}_{\text{out}}^{l-1} \in \mathcal{R}^{N \times d}$ are the hidden representations from the preceding layer, and $\text{self-attn}(\mathbf{X}_{\text{out}}^{l-1})_i \in \mathcal{R}^d$ is the output from the self-attention layer being added to the i -th hidden representation. The adapter layer is typically a small FFN with a low-dimensional bottleneck:

$$\text{adapter}(\mathbf{x}) = \mathbf{W}_2 \sigma(\mathbf{W}_1 \mathbf{x}), \quad (4)$$

³Adapters with a reduction factor of 16, i.e. from 1024 to 64, and the ReLU activation function as implemented in Pfeiffer et al. (2020a)

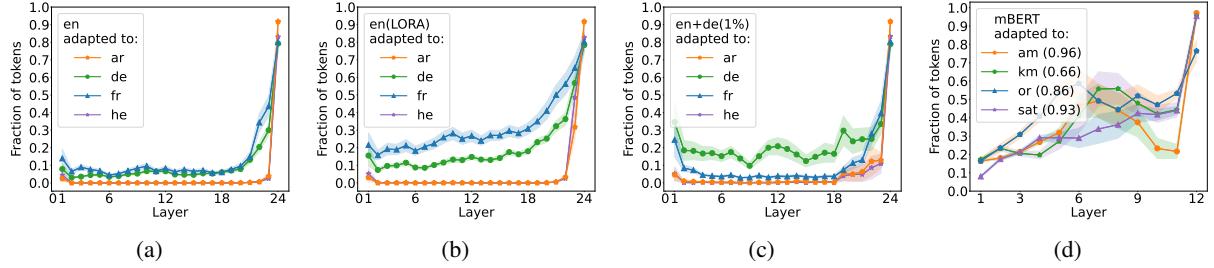


Figure 1: The fraction of target language tokens in the top-10 predicted tokens when projecting hidden layer representations to the vocabulary space. (a) shows results for a model pre-trained only on English, (b) shows results for a model pre-trained only on English, and adapted with LoRA, (c) shows results for a model pre-trained on English and 1% of the German data, (d) shows results for mBERT.

where σ is a non-linear activation function, $\mathbf{W}_1 \in \mathcal{R}^{b \times d}$, $\mathbf{W}_2 \in \mathcal{R}^{d \times b}$ and $b \ll d$. We will refer to this setting as Pfeiffer adapters for the remainder of this paper.

In addition to analyzing Pfeiffer adapters, we extend our analysis to LM adaptation with low-rank adapters (LoRA) (Hu et al., 2021). To ensure comparison with the Pfeiffer adapter setup, we add low-rank adapters only to the FFN layers within each transformer decoder block. We describe and formalize the LoRA setup in more detail in Appendix D.1.

In our analysis, we view each layer (self-attention, feed-forward, and adapter) as reading from and writing to the model’s residual stream (Elhage et al., 2021). Crucially, during adaptation all pre-trained weights, except for the input and output embeddings, are frozen. Training the embeddings in addition to the adapters is necessary, as our original model was pre-trained exclusively on English, and it also leads to improved adaptation performance (Artetxe et al., 2020; Conneau et al., 2020; Yong and Nikoulina, 2022).⁴

Data We used data sourced from Wikipedia for both the pre-training and adaptation stages. Details on the data sizes are provided in Appendix B. For our analysis, we use data sampled from FLORES-101 (Goyal et al., 2022), a dataset of 3k English sentences carefully translated to various languages. We combined the dev and devtest splits for our analysis. In addition, we used the parallel universal dependencies (PUD) treebanks (Zeman et al., 2017) for en, de, and fr for POS analysis, and the multilingual morphosyntactic probing dataset for de, and fr (Acs et al., 2023).

⁴An alternative approach is to use invertible adapters (Pfeiffer et al., 2020b), but it requires that the input and output embeddings are tied, which does not hold in our case.

3 Adapted predictions evolve in the source language

Adapters operate on top of the computation of the frozen pre-trained LM, introducing changes that the subsequent layer does not “expect” since its parameters were not updated during the adapter training. This raises the question of whether the adapted model “thinks” in the distribution of the source language(s) it predominantly saw during pre-training and translates its predictions during adaptation or whether it operates exclusively in the target language it was adapted to.

Experiment To address this question, we inspect the adapted LM predictions across layers via projection to the vocabulary space (Nostalgebraist, 2020; Geva et al., 2022b). We feed examples in each target language into the model and extract the hidden representations created during inference at the last position, from which the next-token prediction is obtained. Then, we project each representation to the vocabulary by multiplying it by the LM head, and apply the softmax function to obtain an intermediate output distribution for every hidden representation. For each example we analyze in which language — the source language the LM was trained on or the target language — the LM constructs its adapted prediction. This is done by taking the top- k tokens with the highest probabilities in each layer, and categorizing them to their respective languages. We perform this classification based on the script for ar and he and based on token statistic for en, de, and fr (see Appendix C for details).

Results Figure 1a shows the portion of predicted tokens from the target language with $k = 10$, across the layers of a monolingual English model adapted to each of the target languages indepen-

dently.⁵ For all target languages, we observe that starting from the second layer and up until the last two layers, the presence of target language tokens remains notably limited, suggesting that the vast majority of the layers promotes tokens in the pre-training language rather in the target language. In contrast, for the last two layers we observe a dramatic increase of target language tokens presence in the top of the output distribution. Figure 1b shows a similar trend when adapting with LoRA instead.

We additionally pre-trained a model on 1% of the German data together with all of the English data. We adapt this model to all target languages and repeat our previous analysis. Figure 1c shows the result. As expected, the fraction of German tokens is now higher, even for the middle layers. Overall, however, we still see the same pattern as before, with most of the predictions being in the source language except for the last two layers.

Generalization to multilingual models We repeat the same experiment but for an existing multilingual LM, namely mBERT (Devlin et al., 2019). We exploit the fact that mBERT has not been trained on Amharic (am), Khmer (km), Odia (or), and Santali (sat) for which we can obtain sufficient data for adaptation. For language identification, we follow the same approach as described above and rely on the script of each language. To adapt mBERT, we first train new tokenizer jointly on all of these new languages and extend mBERT’s vocabulary with the newly created vocabulary. As with our models, we train only the embeddings and adapters and project the hidden representations of every layer to the vocabulary at inference time.

Figure 1d shows that in contrast to the mono- and bilingual LMs, for mBERT we observe a substantial increase in target language tokens already in the middle layers, reaching up to 60% of the top tokens in the projection. We attribute this to the multilingual representation space of mBERT, which potentially helps the model to adapt to new languages. Nonetheless, we again observe that the fraction of target tokens increases dramatically at the final layer.

Conclusion We take these observations as evidence that adapted predictions are evolved in the distribution of the source languages the model saw

⁵We choose $k = 10$ following Geva et al. (2022b). Similar trends were observed for $k = 100, 1000$.

during pre-training, while being shifted to the target language only at the end of the inference pass.

4 The adaptation process is distributed across layers

Our previous findings raise the question of how distributed the adaptation process is, i.e., is it concentrated on specific adapters or distributed across all layers of the model? Asked differently, are only the last adapters important while those at the early layers can be ignored or removed? In the following, we tackle this question by analyzing the magnitude of adapter outputs and the effect of canceling individual and consecutive groups of adapters during inference on the adaptation performance.

4.1 Adapters gradually steer the prediction

Every adapter at layer l contributes an additive update to the evolving residual stream representation.⁶ Here we analyze how pronounced these updates are in terms of their magnitude. To this end, for each target language, we feed the entire FLORES-101 devtest data to our models and we obtain the adapter, feed-forward, and layer outputs for 6500 tokens at randomly chosen positions. Then, we compare the average L2 norm of these representations at every layer. If not stated otherwise, we focus on the Pfeiffer configuration in the following, i.e. we compare $\|\text{adapter}(\mathbf{x}_{\text{ffn}}^{l,N})\|_2$ with $\|\mathbf{x}_{\text{ffn}}^{l,N}\|_2$ and $\|\text{feed-forward}(\mathbf{x}_{\text{attn}}^{l,N})\|_2$.

Figure 2 shows that adapters introduce updates that are substantially smaller in magnitude compared to the residual stream representations. Moreover, the adapter outputs are similar in magnitude to those of the FFN layers, which previously have been shown to introduce gradual changes to the prediction (Geva et al., 2021). While upper-layer adapters introduce updates of larger magnitude compared to early-layer adapters, this increase in norm is also observed for the hidden representations, and thus may not explain the shift towards the target language observed in §3. Interestingly, the norm of adapter outputs in the last layers is larger for Arabic and Hebrew than for German and French, suggesting that larger updates are needed to steer predictions to more distant languages.

⁶ $\text{adapter}(\mathbf{x}_{\text{ffn}}^{l,N})$ in the Pfeiffer configuration and $\text{LoRA}_2(\mathbf{x}_{\text{ffn}_1}^{l,i})$ with LoRA (see Appendix D.1).

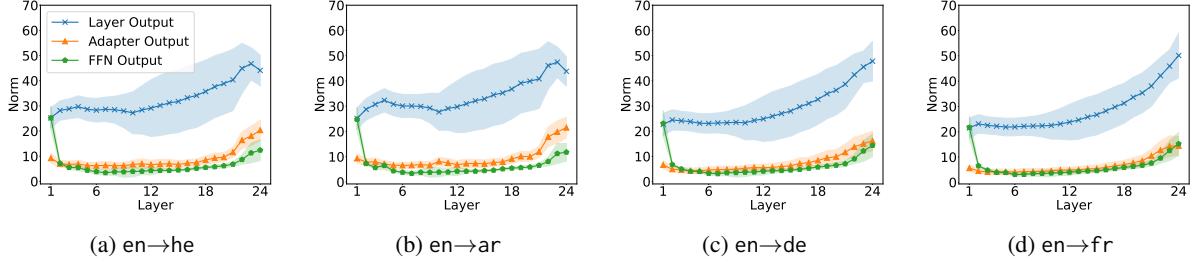


Figure 2: The average L2 norm of the adapter output in comparison to the feed-forward and layer outputs.

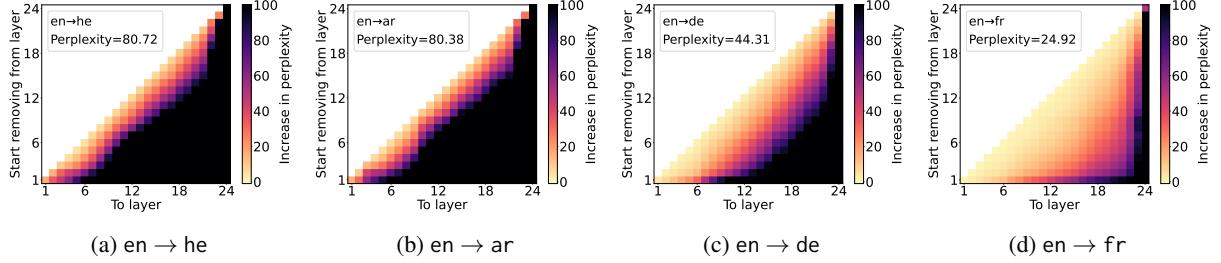


Figure 3: The increase in perplexity when removing adapter layers during inference for models adapted from en to he, ar, de, fr, respectively. To aid visibility, we cap the increase in perplexity at 100.

4.2 Adapters often can be removed with only a small effect on perplexity

To investigate the importance of individual adapter layers, we zero-out the output from single or consecutive adapter layers during inference (similarly to Haviv et al. (2023); Wang et al. (2023)) and measure how this intervention affects perplexity on the held out Wikipedia validation set. Concretely, for every layer l' and any of its succeeding layers $l'' = l' + 1, \dots, L$, we intervene on the inference pass and set the outputs of all the adapters between layers l' and l'' to zero, i.e. $\mathbf{x}_N^{l', \text{out}} \leftarrow \mathbf{x}_N^{l', \text{ffn}} \forall l \in [l', l'']$.

Figures 3a to 3d show the average increase in validation perplexity of our adapted models for every intervention and each target language. We observe consistent results when adapting with LoRA and also when adapting mBERT, a massively multilingual model, and discuss these results in more detail in Appendix E.1. We observe that removing individual layers typically has a minor impact on perplexity, except for the last two layers where perplexity dramatically increase to over 100 across all languages. This effect is consistent with our findings in §3, where we observe the largest increase in target tokens at the last layer.

The increase in perplexity becomes even more pronounced as the number of removed adapters increases and when considering target languages that are more distant from the source language. For example, for most layers, removing three consecutive adapters leads to an increase of up to 25

points perplexity for Arabic and Hebrew compared to less than 10 points for German and French.

This suggests that adapting from English to Hebrew or Arabic is more difficult than adaptation to German or French as individual adapters (and especially small groups of adapters) have a more profound impact on the adaptation performance. Stated differently, we hypothesize that for languages more distant from the source language, adapters have to “do more work”, therefore, removing them leads to larger drops in performance.

5 Two adaptation hypotheses

Our experiments so far have analyzed how LM adapters influence the predictions of the underlying model, making several key observations: 1) adapted LM predictions are evolved in the source language distribution; decoding the output distribution from most hidden layers shows that source language tokens are substantially more pronounced than target language tokens, until reaching the very last layers where the target language abruptly becomes pronounced; 2) the adaptation process is gradual across most layers, as it is possible to skip multiple adapters without decrease in performance, while the last few adapters are crucial for adaptation success; 3) the adaptation of our pretrained model is notably better for French and German than for Hebrew and Arabic. This is reflected in the larger norm of the adapter updates for these two languages and the bigger impact on perplexity

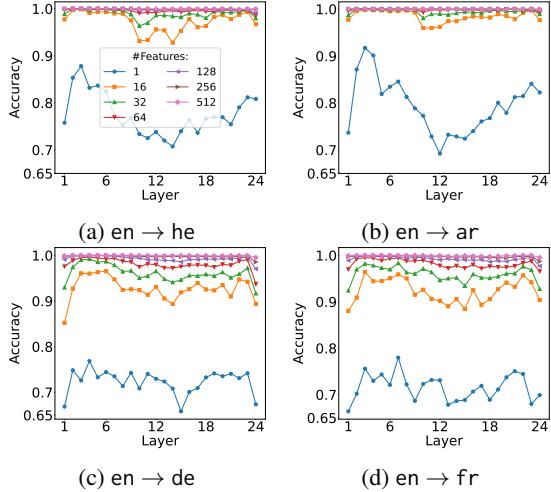


Figure 4: Sparse probing classifiers can detect language adaptation with high accuracy.

when removing chunks of adapters. We attribute this finding to the fact that English shares a linguistic and etymological connection with German and French. In contrast, for Arabic and Hebrew, which utilize non-Latin scripts, adaptation appears to be more difficult due to the difference in linguistic and orthographic characteristics.

From these findings, we conclude that the adaptation process during the forward pass is distributed across all layers, with each individual layer—except the last few layers—making gradual updates to the overall adaptation. In what follows, we set out to investigate two alternative hypotheses that could explain the interplay between the adapter updates and the underlying prediction space.

Hypothesis 1: Adapters operate in an isolated subspace We hypothesize that adapters operate in a specific subspace of the model’s residual stream which is not utilized by the underlying model, leaving most of the representation space untouched. This isolated representation space then becomes pronounced in the last few layers, leading to the prediction of tokens in the target language.

Hypothesis 2: Adapters operate on top of the model’s representation space An alternative hypothesis is that adapters operate on top of the structure in the underlying model’s representation space, preserving its overall structure while gradually pushing representations closer to the embedding space of the target language.

5.1 Do adapters operate in an isolated subspace of the residual stream?

We approach the first hypothesis via a sparse probing experiment. Specifically, we train probing classifiers to predict whether a given layer output has been adapted or not. In a first step, we follow Gurnee et al. (2023) and identify critical features for adaptation using the maximum mean differences (MMD) algorithm which ranks features based on their importance for distinguishing between adapted and non-adapted features (a detailed description is provided in Appendix E.2). Next, we take the top-k most important features identified by MMD and use them as input features for a binary logistic regression classifier which we train to predict whether or not a representation has been adapted at that layer.

Figure 4 shows that for various levels of probe sparsity, a linear probing classifier can predict with high accuracy whether a given hidden representation has been updated by a specific adapter or not. Interestingly, across all levels of sparsity, adaptation is easier to predict for Hebrew and Arabic than for German and French. This provides further evidence that adaptation is more pronounced on the underlying model representations for these languages.

Next, we intervene on the features identified by MMD to investigate the extend to which these features are involved in predicting tokens from the target language. We zero-out individual dimensions of the adapter outputs which correspond to the most important features identified by MMD and compare the target language perplexity on our adaptation validation split before and after intervention.⁷ As baselines, we intervene on the least important features identified by MMD as well as on randomly selected features.

Figure 5 depicts the adapted model perplexity for increasing number of intervened features. While intervening on the most important features indeed leads to a dramatic increase in perplexity on all target languages, intervening on the least important or even on random features also leads to a substantial increase in perplexity.

Refuting Hypothesis 1 From the findings above we refute the hypothesis that adapters operate in an isolated subspace of the residual stream of the underlying model, as even intervening on random

⁷Replacing the value of individual dimensions by the average of other feature directions leads to very similar results (see Appendix E.3).

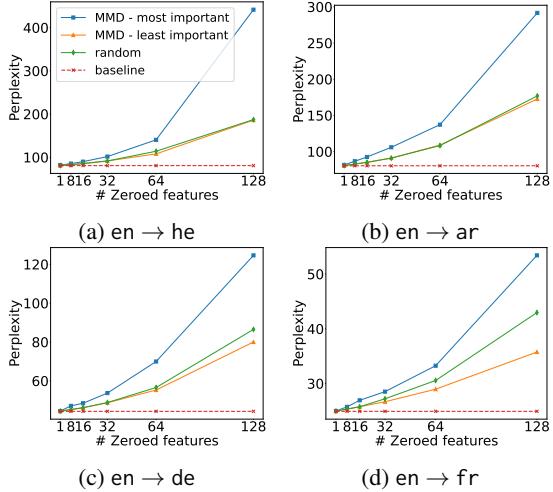


Figure 5: Adapted model perplexity after intervention.

features leads to a substantial increase in perplexity. We note that adapters might still operate in a subspace, however, our findings show that they do not operate in an isolated subspace which is uniquely used by the adapters.

5.2 Adapters largely preserve the structure of the underlying model

To approach the second hypothesis, we compare the structure of the residual stream in adapted versus non-adapted predictions with respect to different linguistic properties. Specifically, we analyze the structure corresponding to part of speech (adposition/determiner/noun/verb), verb tense (past/present), and grammatical number (singular/plural). Part of speech (POS) labels were obtained from PUD (Zeman et al., 2017) and data for verb tense and grammatical number were obtained from the data released by Acs et al. (2023).

Given a property, we first obtain token representations from the monolingual model for 1100 inputs in English which correspond to different values of this property. For every layer l , we run PCA on the hidden representations and create a projection matrix $P^l \in \mathcal{R}^{2 \times d}$ consisting of the first two principal components. Next, we apply the projection matrix obtained from the English representations to layer output representations from the adapted models for German and French inputs, again focusing on representations of tokens with different values of the property.

Figure 6 visualizes the projection results for different layers of the models for POS. Results for additional layers, with LoRA, as well as tense and number all show very consistent trends and are dis-

cussed in Appendix E.4. Focusing on the English representation space first (first column), the POS structure is clearly visible across all layers. Interestingly, using the same projection matrix derived from the English layer outputs, reveals a highly similar structure in the adapted representation space for German and French (second and third columns).

Figure 7 provides an alternative way to view the results, showing for every layer the cosine similarity between the first two principal components obtained from applying PCA to the English layer outputs and each of the German and French layer outputs, separately. For almost all layers, we observe a very high alignment (absolute cosine similarity ≈ 0.6) between the principal components (similar plots for number and tense are provided in Appendix E.4).

Conclusion Overall, we take these observations as evidence that the modifications of the adapter layers operate on top of the already existing structure in the representation space of the pre-trained model. This also means that the adapters are constrained by this space and making more drastic changes will require more adaptation (more adapters/larger-in-magnitude updates), which is consistent with our findings in previous sections (§3, §4).

6 Related work

Language adaptation Various efforts have been aiming to extend the capabilities of pre-trained LMs to previously unseen languages (Chau et al., 2020; Pfeiffer et al., 2021b; Yong and Nikoulina, 2022; Alabi et al., 2022; ImaniGooghari et al., 2023, *inter alia*). While prior approaches have largely relied on continued or adaptive pre-training which updates all the model’s parameters (Chi et al., 2021; Alabi et al., 2022; ImaniGooghari et al., 2023), more recent approaches rely on parameter efficient methods (Houlsby et al., 2019; Pfeiffer et al., 2020b; Marchisio et al., 2023) and have been shown to be effective across many languages. Our work seeks to provide a better understanding of the adaptation process via adapters, which could potentially lead to concrete ways to improve it.

How transformer-based LMs build predictions

Several prior works have studied the evolution of predictions in transformer-based LMs (Voita et al., 2019; Tenney et al., 2019; NostalgiaBraist, 2020; Geva et al., 2022a; Din et al., 2023; Belrose et al.,

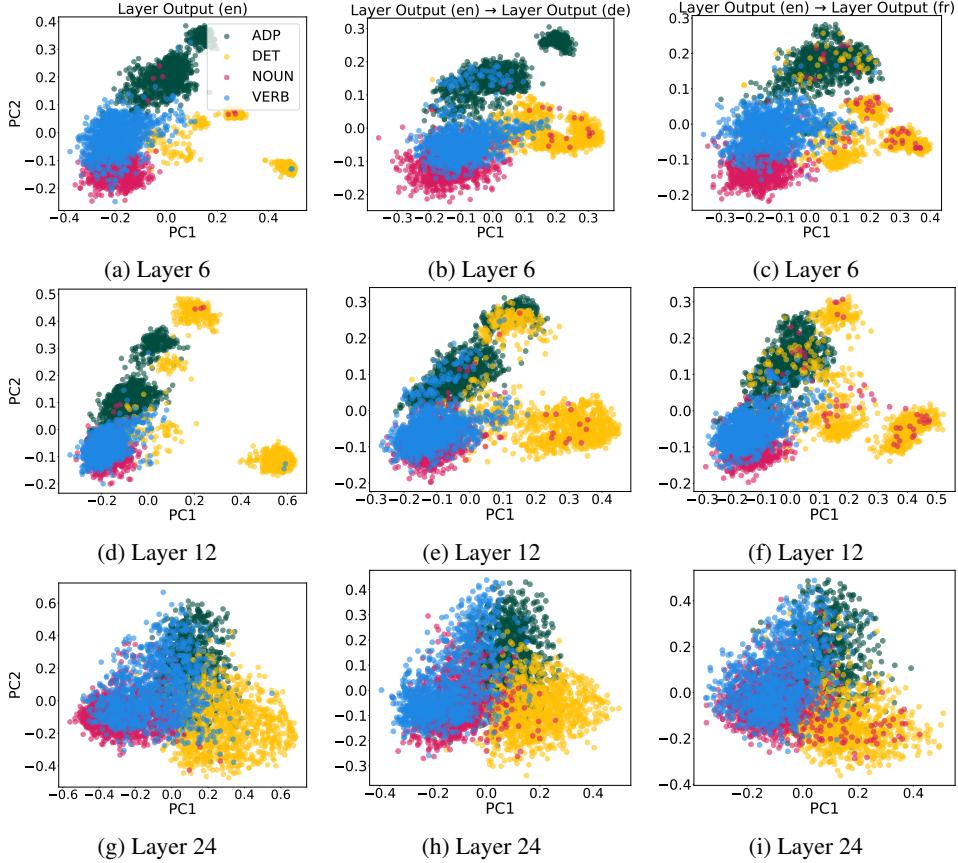


Figure 6: 2D projections for tokens with different POS of the model pre-trained on English (first column) and the adapted models trained on German (second column) and French (third column) at various layers. In all three cases, the projection matrix is computed via PCA on the English representations only.

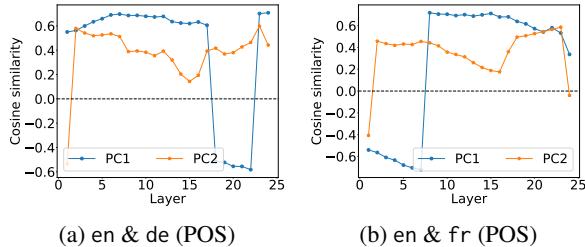


Figure 7: Cosine similarity between the first two principal components of residual stream. PCA is computed on token representations with different POS on English and the target languages separately.

2023; Ferrando et al., 2023). Specifically, Geva et al. (2021, 2022b) showed that FFN layers in transformers gradually build predictions by promoting concepts that are interpretable in the vocabulary space. Unlike these works, here we analyze the evolution of *adapted predictions*, providing a unique perspective on how adapters steer the frozen LM prediction process.

Analyzing multilingual language models Several works have analyzed multilingual LMs with

special focus on the representations of these models. Some of these works have shown that these models learn language-agnostic representations which are necessary for cross-lingual transfer (Pires et al., 2019; Libovický et al., 2020; Muller et al., 2021; Zhao et al., 2021; Xie et al., 2022; Chang et al., 2022; Abdullah et al., 2023).

Analyzing adapters Only few works have analyzed the specific role of adapters during adaptation. Rücklé et al. (2021) proposed Adapter-Drop, which involves either dynamically dropping adapters during training to enhance robustness or to speed up inference. He et al. (2021) showed that adapter-based fine-tuning results in representations with less deviation from the original model at each layer, leading to better training stability and generalization without forgetting compared to full fine-tuning. Recently, Kunz and Holmström (2024) investigate the impact of target language adapter in zero-shot cross-lingual transfer, revealing that language adapters have only a minor impact on downstream performance.

7 Conclusion and Discussion

We experiment with language adaptation applied to mono-, bi-, and multilingual pre-trained LMs and study how language adapters interact with the underlying model. We show that adapted LM predictions are mostly evolved in the distribution of the source language(s) the model saw during pre-training and that the adaptation is gradually happening on top of the existing representation structure of the underlying models. We not only provide a unique perspective on the inner-working of language adaptation but our findings also open up several interesting avenues for future work on language adaptation.

Future research on designing more efficient adaptation approaches could build on our findings on the relationship between the “ease of adaptation” and source-target language similarity, to automatically reduce the number of adapters during inference and to study the transfer of adapters across languages. Our insights on the alignment between adapters and the underlying representation space could be extended to investigate the extend to which this alignment restricts the amount of adaptation possible informing studies on alternative ways of adaptation which are less constrained by the existing structure in the pre-trained model, which might lead to better performance on languages that are more distant from the source language.

Limitations

Most of our experiments are conducted on models we trained from scratch using a relatively small corpus. These models are easier to analyze compared to existing multilingual models, which allows us to perform analyses in controlled settings. Extending these experiments to existing multilingual models, such as mGPT, is a valuable non-trivial effort to pursue, which we leave for future work.

Our experiments that show how adapted predictions are evolved in the distribution of source language(s) (Section 3) rely on the recent method of projecting hidden representations to the LM vocabulary space, which only provides an approximation to the information encoded in intermediate representations (Din et al., 2023). Nonetheless, the low rate of target language tokens is unlikely to be explained only by this, as it is also low in the middle-upper layers where approximation is better (Geva et al., 2022b, 2023; Merullo et al., 2023; Hendel et al., 2023).

Lastly, we focus on 8 specific target languages for our experiments. We leave it to future work to extend our analysis beyond these languages.

Acknowledgements

We thank Badr Abdullah, Vagrant Gautam, Jonathan Hertzig, Shauli Ravfogel, and Julius Steuer for their valuable feedback and discussions. Jesujoba O. Alabi was supported by the BMBF’s (German Federal Ministry of Education and Research) SLIK project under the grant 01IS22015C.

References

- Badr M. Abdullah, Mohammed Maqsood Shaik, and Dietrich Klakow. 2023. [On the nature of discrete speech representations in multilingual self-supervised models](#). In *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 159–161, Dubrovnik, Croatia. Association for Computational Linguistics.
- Judit Acs, Endre Hamerlik, Roy Schwartz, Noah A. Smith, and Andras Kornai. 2023. [Morphosyntactic probing of multilingual bert models](#). *Natural Language Engineering*, page 1–40.
- Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. [Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. [On the cross-lingual transferability of monolingual representations](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.
- Giuseppe Attardi. 2015. Wikiextractor. <https://github.com/attardi/wikiextractor>.
- Nora Belrose, Zach Furman, Logan Smith, Danny Hallawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. [Eliciting latent predictions from transformers with the tuned lens](#). *arXiv*.
- Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. [GPT-NeoX-20B: An open-source autoregressive language model](#). In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language*

- Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.
- Laurie Burchell, Alexandra Birch, Nikolay Bogoychev, and Kenneth Heafield. 2023. [An open dataset and model for language identification](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 865–879, Toronto, Canada. Association for Computational Linguistics.
- Tyler Chang, Zhuowen Tu, and Benjamin Bergen. 2022. [The geometry of multilingual language model representations](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 119–136, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. [Parsing with multilingual BERT, a small corpus, and a small treebank](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1324–1334, Online. Association for Computational Linguistics.
- Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. [InfoXLM: An information-theoretic framework for cross-lingual language model pre-training](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. 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, Online. Association for Computational Linguistics.
- 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, Minneapolis, Minnesota. Association for Computational Linguistics.
- Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. 2023. [Jump to conclusions: Short-cutting transformers with linear transformations](#). *arXiv*.
- Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conery, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. [A mathematical framework for transformer circuits](#). *Transformer Circuits Thread*.
- Javier Ferrando, Gerard I. Gállego, Ioannis Tsiamas, and Marta R. Costa-jussà. 2023. [Explaining how transformers use context to build predictions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5513, Toronto, Canada. Association for Computational Linguistics.
- Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. [Dissecting recall of factual associations in auto-regressive language models](#).
- Mor Geva, Avi Caciularu, Guy Dar, Paul Roit, Shoval Sadde, Micah Shlain, Bar Tamir, and Yoav Goldberg. 2022a. [LM-debugger: An interactive tool for inspection and intervention in transformer-based language models](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 12–21, Abu Dhabi, UAE. Association for Computational Linguistics.
- Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022b. [Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. [Transformer feed-forward layers are key-value memories](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc’Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. [The Flores-101 evaluation benchmark for low-resource and multilingual machine translation](#). *Transactions of the Association for Computational Linguistics*, 10:522–538.
- Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. [Finding neurons in a haystack: Case studies with sparse probing](#). *arXiv preprint arXiv:2305.01610*.
- Adi Haviv, Ido Cohen, Jacob Gidron, Roei Schuster, Yoav Goldberg, and Mor Geva. 2023. [Understanding transformer memorization recall through idioms](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 248–264, Dubrovnik, Croatia. Association for Computational Linguistics.

- Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. 2021. *On the effectiveness of adapter-based tuning for pretrained language model adaptation*. 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 2208–2222, Online. Association for Computational Linguistics.
- Roei Hendel, Mor Geva, and Amir Globerson. 2023. *In-context learning creates task vectors*. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9318–9333, Singapore. Association for Computational Linguistics.
- 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 *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. *Lora: Low-rank adaptation of large language models*.
- Ayyoob ImaniGooghari, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. *Glot500: Scaling multilingual corpora and language models to 500 languages*. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.
- Jenny Kunz and Oskar Holmström. 2024. *The impact of language adapters in cross-lingual transfer for nlu*.
- Anne Lauscher, Olga Majewska, Leonardo F. R. Ribeiro, Iryna Gurevych, Nikolai Rozanov, and Goran Glavaš. 2020. *Common sense or world knowledge? investigating adapter-based knowledge injection into pretrained transformers*. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 43–49, Online. Association for Computational Linguistics.
- Jaeseong Lee, Seung-won Hwang, and Taesup Kim. 2022. *FAD-X: Fusing adapters for cross-lingual transfer to low-resource languages*. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 57–64, Online only. Association for Computational Linguistics.
- Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. 2020. *On the language neutrality of pre-trained multilingual representations*. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1663–1674, Online. Association for Computational Linguistics.
- Kelly Marchisio, Patrick Lewis, Yihong Chen, and Mikel Artetxe. 2023. *Mini-model adaptation: Efficiently extending pretrained models to new languages via aligned shallow training*. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5474–5490, Toronto, Canada. Association for Computational Linguistics.
- Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2023. *A mechanism for solving relational tasks in transformer language models*.
- Benjamin Muller, Yanai Elazar, Benoît Sagot, and Djamé Seddah. 2021. *First align, then predict: Understanding the cross-lingual ability of multilingual BERT*. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2214–2231, Online. Association for Computational Linguistics.
- Nostalgebraist. 2020. *Interpreting GPT: The logit lens*.
- Marinela Parović, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2022. *BAD-X: Bilingual adapters improve zero-shot cross-lingual transfer*. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1791–1799, Seattle, United States. 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: System Demonstrations*, pages 46–54.
- 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 (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.
- 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.
- Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. **How multilingual is multilingual BERT?** In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.
- 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.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. **BERT rediscovers the classical NLP pipeline.** In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Elena Voita, Rico Sennrich, and Ivan Titov. 2019. **The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives.** 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 4396–4406, Hong Kong, China. Association for Computational Linguistics.
- Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2019. **Neural machine translation with byte-level subwords.** *arXiv*.
- Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. **Interpretability in the wild: A circuit for indirect object identification in GPT-2 small.** In *The Eleventh International Conference on Learning Representations*.
- Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Dixin 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.
- Zhihui Xie, Handong Zhao, Tong Yu, and Shuai Li. 2022. **Discovering low-rank subspaces for language-agnostic multilingual representations.** In *Proceedings of the 2022 Conference on Empirical Methods* *in Natural Language Processing*, pages 5617–5633, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Zheng-Xin Yong and Vassilina Nikoulina. 2022. **Adapting bigscience multilingual model to unseen languages.**
- Daniel Zeman, Martin Popel, Milan Straka, Jan Hajíč, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajíč jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisoroj, and Josie Li. 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*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.
- Wei Zhao, Steffen Eger, Johannes Bjerva, and Isabelle Augenstein. 2021. **Inducing language-agnostic multilingual representations.** In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 229–240, Online. Association for Computational Linguistics.

A Details on model training

Tokenization and training data We pretrain and adapt our models using data sourced from Wikipedia using WikiExtractor (Attardi, 2015). For tokenization, we use the BPE (Byte Pair Encoding) tokenizer developed by (Wang et al., 2019). Given the noisy nature of webcrawled data such as Wikipedia, we filtered the collected data using the OpenLID (Burchell et al., 2023) language identification model. To train the tokenizer, we sampled 4M sentences from each language’s monolingual data. Thus, in total, we obtained 20M sentences and trained a single tokenizer on data of all languages with using a vocabulary of size 250K. Given this pretrained tokenizer, we uniformly sampled sentences based on the count of tokens from each language’s monolingual data and used the resulting data for pre-training and adaptation.

Adapter training The adapters were trained for 150K steps. We evaluated the model’s on the validation monolingual data every 5000 steps and selected the model (adapter) that yielded the lowest validation perplexity.

B Details on pre-training and adaptation token statistics

Table 1 shows the number of sentences and tokens used for both training and adaptation. In our controlled experiments, Hebrew had the least amount of available monolingual data on Wikipedia. Therefore, to ensure equitable representation across all languages, we opted to sample an equal amount of tokens from Wikipedia for the other languages. Specifically, given the approximately 145 million tokens in Hebrew, we sampled an equal number of tokens from the other languages’ monolingual data. Additionally, 2% of the data was reserved as a development set, with the remainder used for training or adaptation purposes.

We followed a similar approach for the mBERT experiment, where we sampled 4 millions tokens from all languages, and reserved 5% as the development set for each language.

C Details of language token identification

In Section 3, we mention that for French and German, language identification is performed by looking at language statistics. Considering the token counts in three languages (including English) based on the Latin script. We determine the predominant

Language	Code	# Sentences	# Tokens	
			Train	Dev
Arabic	ar	4,001,963	142,492,237	2,908,012
German	de	2,148,944	142,492,054	2,908,064
French	fr	2,187,922	142,491,947	2,908,244
English	en	1,742,844	142,492,202	2,908,014
Hebrew	he	3,078,001	142,492,172	2,908,062
Amharic	am	80,060	3,869,262	203,101
Khmer	km	152,375	3,869,233	203,008
Odia	or	124,900	3,869,209	203,112
Santali	sat	78,887	3,869,077	203,161

Table 1: Pre-training and adaptation data sizes.

language for any randomly selected token based on its frequency. Once we identify the language with the highest token count for that particular token, we calculate the ratio of its occurrences in the English corpus to the occurrences in the identified language. If this ratio is equal to or less than a specified threshold, which we have set at 0.7 through manual inspection of the resulting classification, we classify the token as belonging to that language.

For mBERT, we used basic script identification just as for Arabic and Hebrew, Ge’ez, Khmer, Odia, and Ol Chiki have distinct scripts that can be easily identified.

D Details on mBERT’s language converge

In order to determine the languages to include in the mBERT experiment, we sought languages with scripts not adequately represented in mBERT. We opted for several languages from FLORES, particularly those using scripts distinct from those covered by mBERT. Table 2 shows four languages that we finally selected, their scripts, and the unknown token rates (UNK rates) according to mBERT’s tokenizer when evaluated on each language’s FLORES dev split. The UNK rate confirms that these languages were not adequately covered by mBERT, with Amharic and Santali having the highest values.

Language	Code	Script	UNK rate
Amharic	am	Ge’ez	0.956
Khmer	km	Khmer	0.664
Odia	or	Oriya	0.858
Santali	sat	Ol Chiki	0.933

Table 2: The langauges not covered in mBERT and their UNK rate when FLORES dev splits for these languages were tokenized.

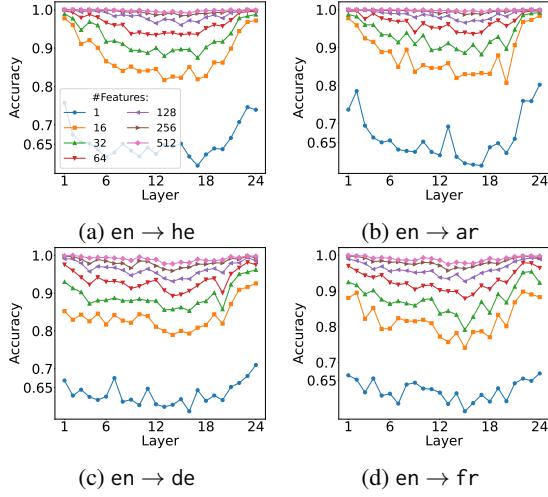


Figure 8: Sparse probing classifiers can detect language adaptation with high accuracy (**case 2**).

D.1 LoRA Adapters

We add low-rank adapters (LoRA) to the FFN sub-layers within each decoder block. With LoRA, the forward-pass of a single layer can be formalized as follows:

$$\mathbf{x}_{\text{attn}}^{l,i} = \mathbf{x}_{\text{out}}^{l-1,i} + \text{self-attn}(\mathbf{X}_{\text{out}}^{l-1})_i \quad (5)$$

$$\mathbf{x}_{\text{ffn}_1}^{l,i} = \text{FFN}_1(\mathbf{x}_{\text{attn}}^{l,i}) + \text{LoRA}_1(\mathbf{x}_{\text{attn}}^{l,i}) \quad (6)$$

$$\mathbf{x}_{\text{ffn}_2}^{l,i} = \text{FFN}_2(\mathbf{x}_{\text{ffn}_1}^{l,i}) + \text{LoRA}_2(\mathbf{x}_{\text{ffn}_1}^{l,i}) \quad (7)$$

$$\mathbf{x}_{\text{out}}^{l,i} = \mathbf{x}_{\text{attn}}^{l,i} + \mathbf{x}_{\text{ffn}_2}^{l,i} \quad (8)$$

Where FFN_1 and FFN_2 are the feed forward sub-layers within the decoder block. Unlike the Pfeiffer adapters, LoRA uses rank decomposition matrices without non-linearity between them:

$$\text{LoRA}(\mathbf{x}) = \mathbf{W}_2(\mathbf{W}_1 \mathbf{x}), \quad (9)$$

where \mathbf{W}_1 and \mathbf{W}_2 are trainable parameters, and LoRA_1 and LoRA_2 have distinct parameters. During adaptation, we only update the parameters of the LoRA and the embedding layer while keeping all other parameters frozen.

E Additional results

E.1 Effect of removing adapters on perplexity

Figure 9 present results for the increase in perplexity when dropping individual or chunks of adapters of mBERT during inference. Similar to the results in Figure 3, we observe little impact on perplexity when removing individual adapters. However, when removing larger consecutive blocks of adapters, perplexity increases for all languages. Compared to Figure 3, the increase in perplexity is

generally less dramatic, which we attribute to the fact that mBERT is a highly optimized multilingual model as well as the fact that masked language modeling is generally an easier task than causal language modeling.

Figure 10 shows result for the same analysis using LoRA. The result are highly consistent with the Pfeiffer adapter setup. Removing individual adapters has little impact on perplexity. However, removing larger consecutive chunks of adapters affects perplexity considerably.

E.2 Details on identifying features critical for adaptation

By design, adapters modify outputs at each layer by adding to the residual stream. Building on our previous observations, we hypothesize that adapters function within a specific subspace of the residual stream. In the following, our objective is to identify the subspace used by adapters at each layer.

We approach this via a sparse probing experiment, where we train probing classifiers to predict whether a given layer output has been adapted or not. We collect positive and negative examples by feeding sequences of each target language to the adapted models and select 6,000 tokens randomly per layer. We considered two cases, and for both cases, for the positive examples, we kept adapters in place at every layer. For the first case (case 1), for the negative examples at layer l , we remove adapters at all layers, including l itself. And secondly (case 2), for the negative examples at layer l , we keep adapters at all previous layers but remove the adapter at layer l itself.

Given these representations, we follow [Gurnee et al. \(2023\)](#) and use the scoring based maximum mean difference (MMD) algorithm to score each of the 1024 features of the residual stream based on the absolute mean difference between the examples from the positive and negative class

$$s_j^l = \frac{1}{|P|} \sum_{i=1}^P X_{ij}^l - \frac{1}{|N|} \sum_{i=1}^N X_{ij}^l, \quad (10)$$

where l is the layer index, j is a single feature dimension, and P and N are the number of positive and negative examples, respectively.

After scoring, we select the top- k ranked features for $k \in \{1, 8, 16, 32, 64, 128, 256, 512\}$ and train a logistic regression probe to discriminate between the positive and negative examples.

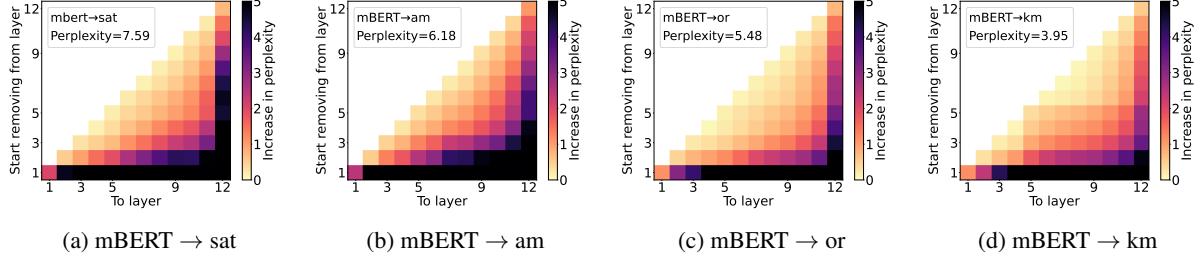


Figure 9: Difference in perplexity after removing some adapters

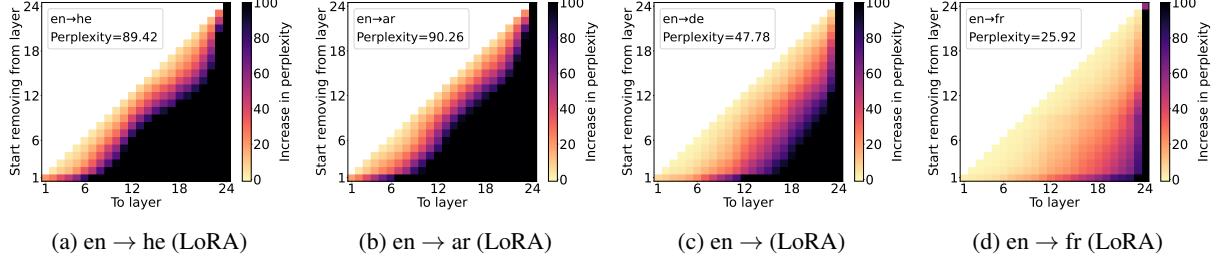


Figure 10: Difference in perplexity after removing some adapters with LoRA

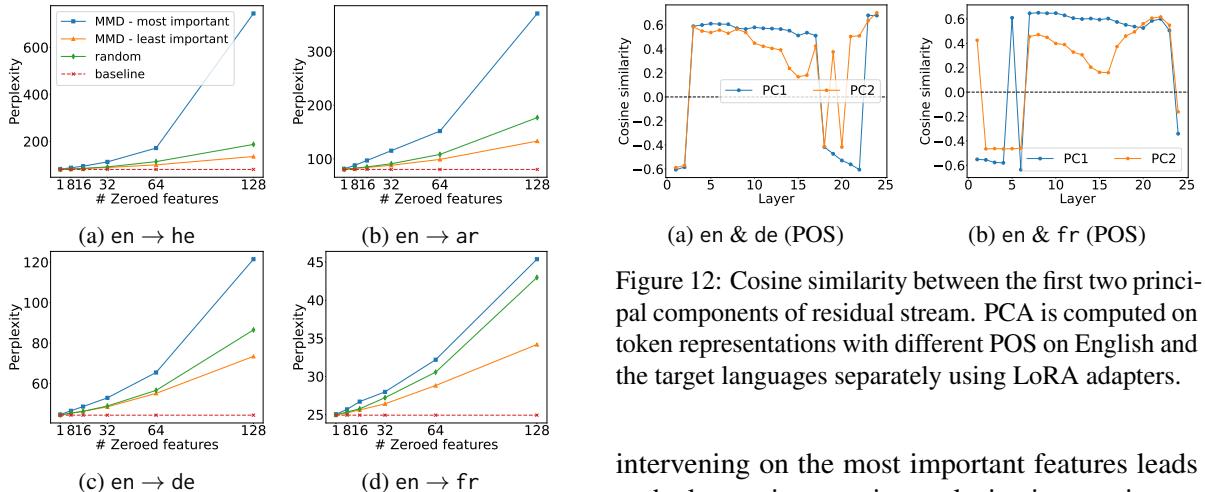


Figure 11: Adapted model perplexity after intervention (case 2).

Figures 4 and 8 show the probing accuracies from the two cases described above respectively. The results show that sparse probing classifiers can detect adaptation with high accuracy.

E.3 More results on intervening on critical features increases perplexity

Instead of zeroing out the adapter representation before updating the residual stream we experiment with an alternative intervention which replaces the features to be intervened on by the average values of the remaining features. We compare the target language validation perplexity after the intervention in Figure 13. Our results show that while

intervening on the most important features leads to the largest increase in perplexity, intervening on the least important or randomly selection similarly leads to a considerable increase in perplexity.

E.4 More results on adapters largely preserve the structure of the underlying model

Part of speech To evaluate whether adapters operate on top of the already existing structure in the representation space, we analyze the structure corresponding to POS (ADP/DET/ NOUN/VERB) for the adapted and non-adapted models. For every layer, we run PCA on the hidden representations for English tokens and create a projection matrix consisting of the first two principal components. We apply the projection matrix obtained from the English representations to layer output representations of German and French inputs with different POS tags. Figure 15 shows the result of this pro-

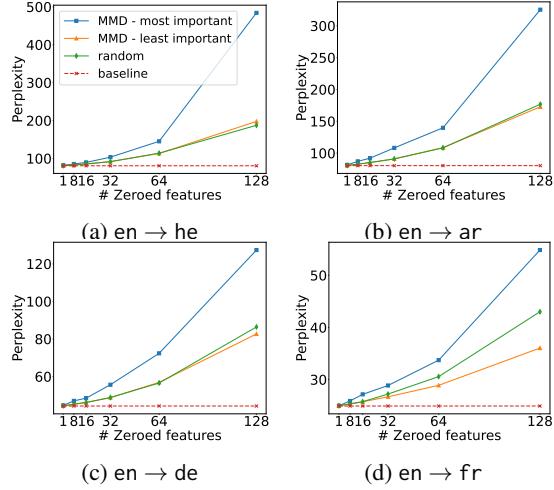


Figure 13: Adapted model perplexity after using the average of non-zero features for intervention.

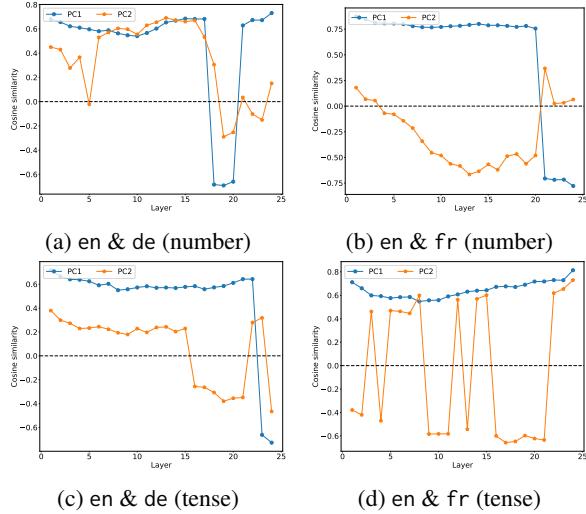


Figure 14: Cosine similarity between the first two principal components of residual stream. PCA is computed on token representations with different number or tense on English and the target languages separately.

jection for several layers. Figure 16 shows highly similar results when using LoRA. Furthermore, Figure 12 shows the cosine similarity between the first two principal components computed on the English and target language representations from the LM adapted with LoRA. Similar to the result from the classical adapters presented in Figure 7, we observe a very high alignment (absolute cosine similarity ≈ 0.6) between the principal components.

Tense and number In addition analyzing the structure corresponding to POS, we also examine more nuanced linguistic features, such as verb tense (present and past) and noun number (singular and plural). Following a similar methodology to the POS experiment, we conduct a principal compo-

nent analysis (PCA) on the hidden representations of English tokens and applied the resulting projection matrix to the output representations of German and French inputs. Figures 17 and 18 shows the results from this experiment.

In addition, Figure 14 shows the cosine similarity between the first two principal components computed on the English and target language representations separately. As with POS, we observe a large alignment between the principal components, providing further evidence for our hypothesis that adapters mostly operate on top of the existing structure of the pre-trained model.

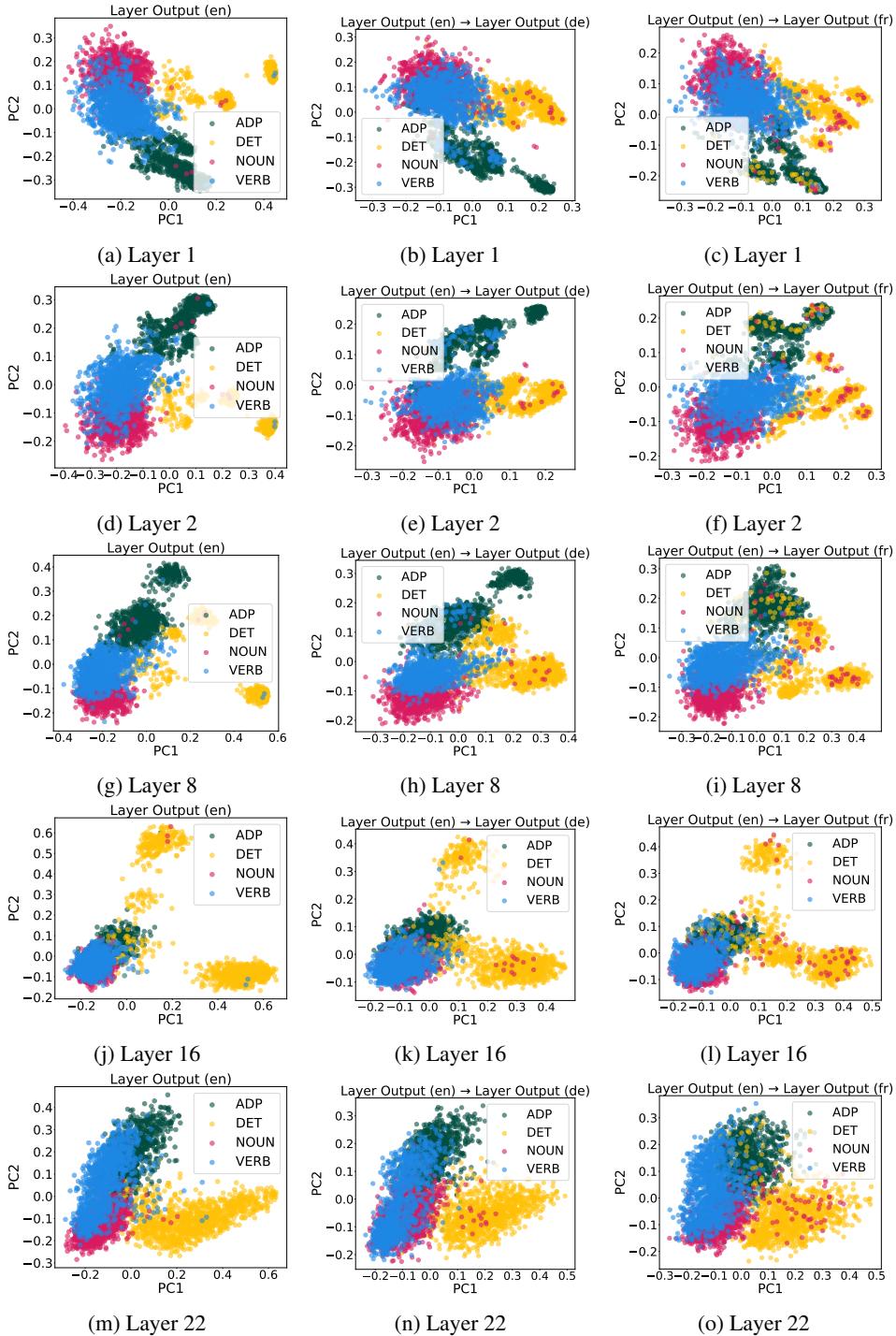


Figure 15: 2D projections for tokens with different POS of the model pre-trained on English (first column) and the adapted models trained on German (second column) and French (third column) at various layers. In all three cases, the projection matrix is computed via PCA on the English representations only.

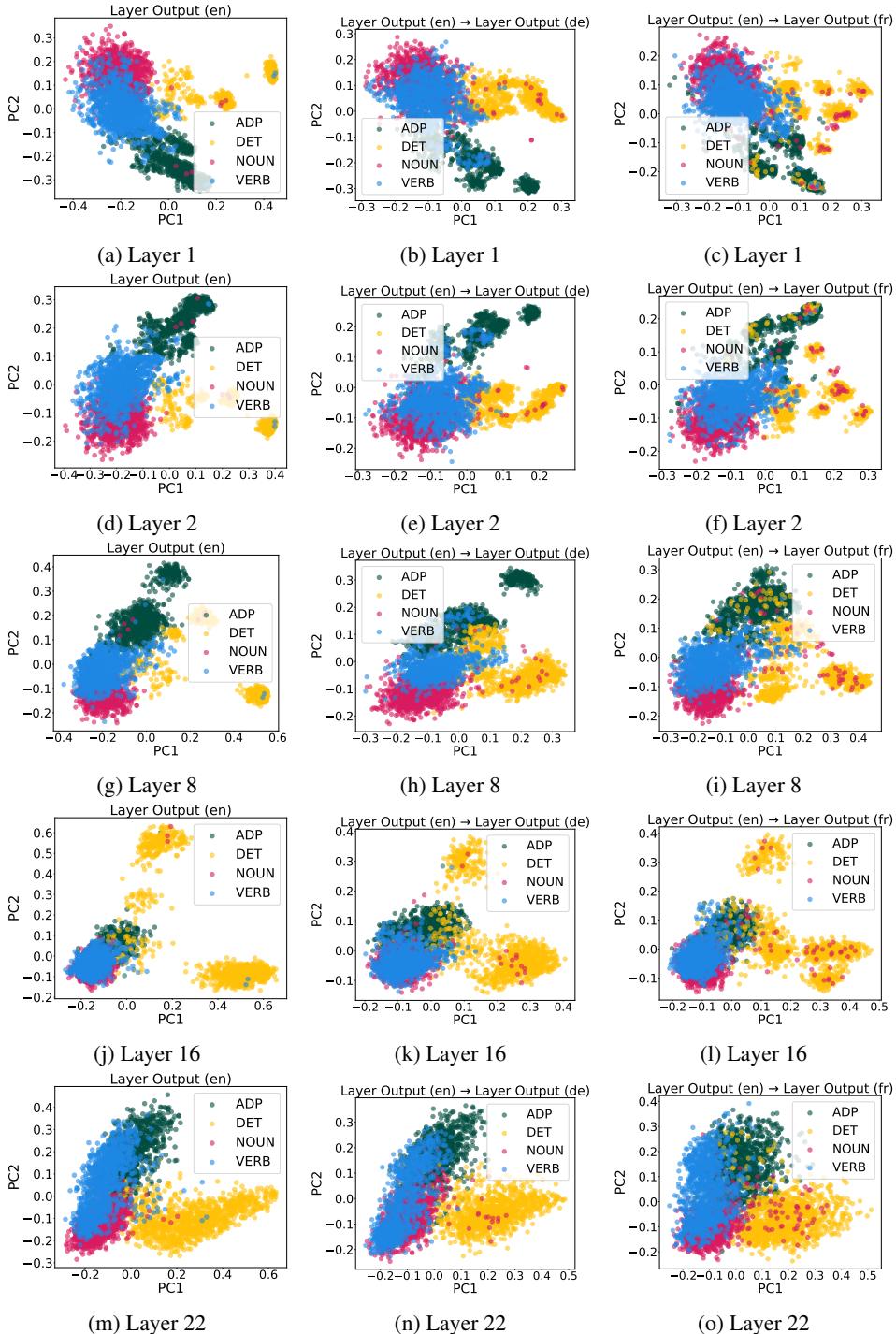


Figure 16: 2D projections for tokens with different POS of the model pre-trained on English (first column) and the adapted models with LoRA trained on German (second column) and French (third column) at various layers. In all three cases, the projection matrix is computed via PCA on the English representations only.

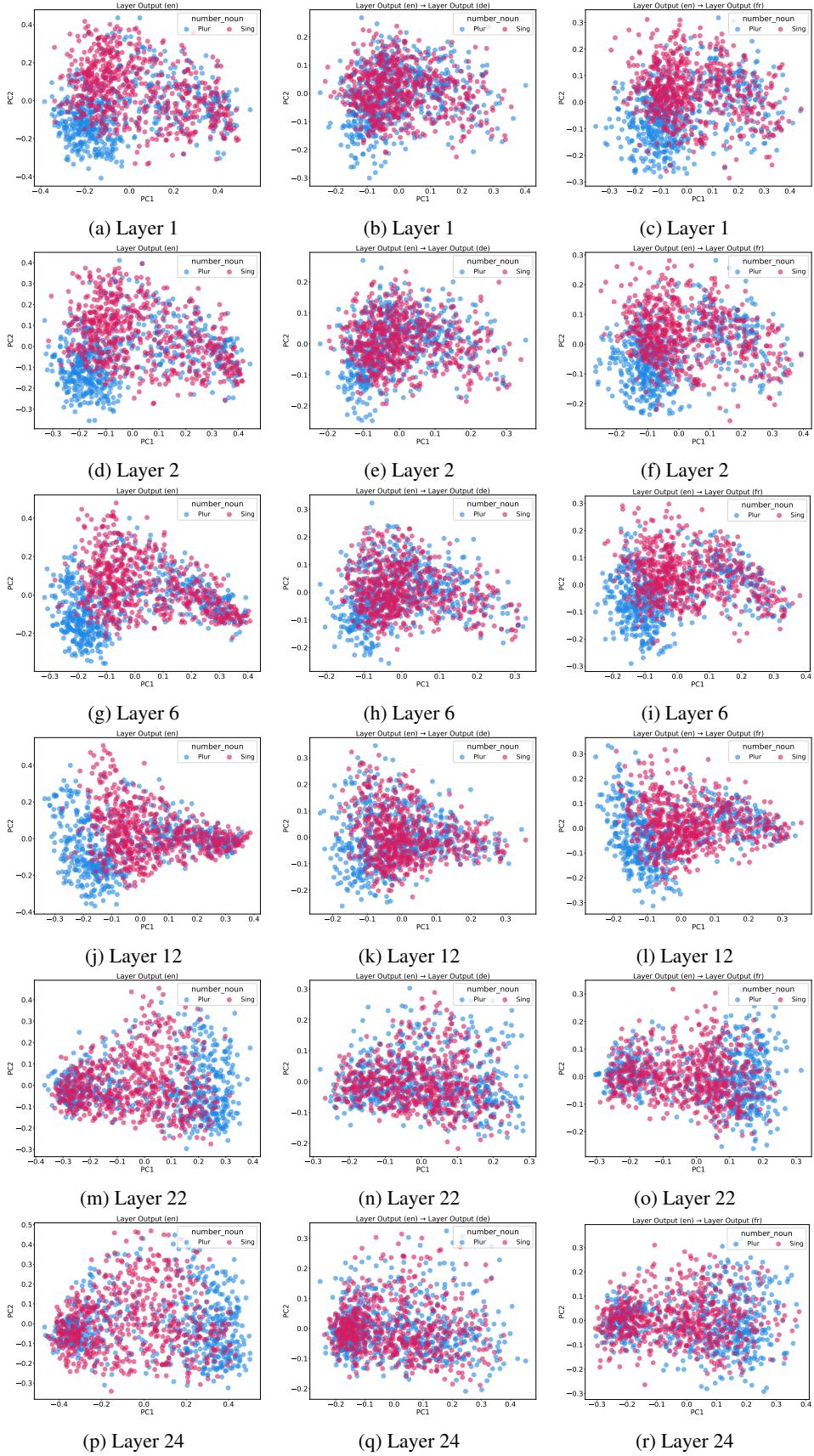


Figure 17: 2D projections for tokens with different number (plural vs. singular) of the model pre-trained on English (first column) and the adapted models trained on German (second column) and French (third column) at various layers. In all three cases, the projection matrix is computed via PCA on the English representations only.

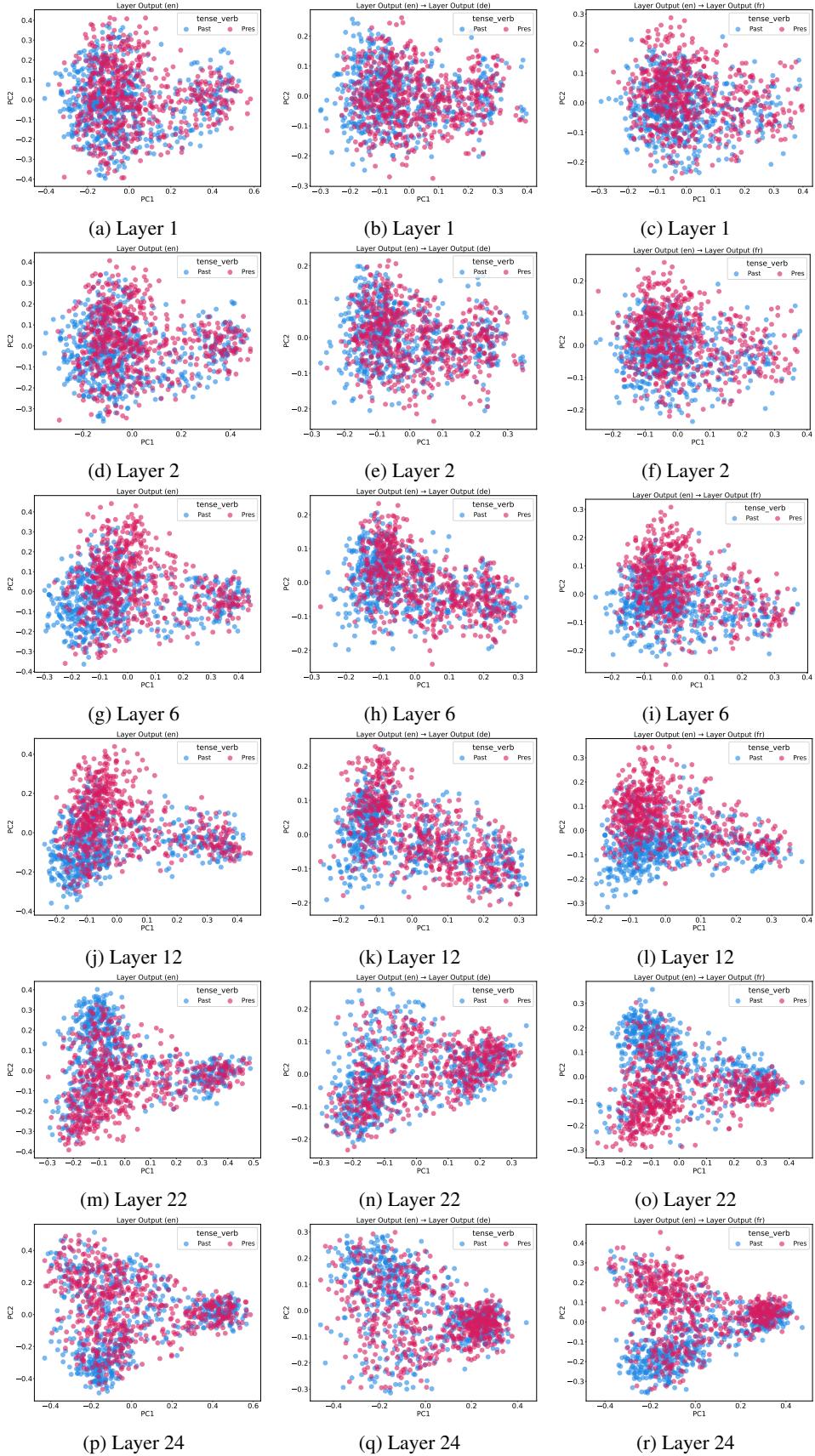


Figure 18: 2D projections for tokens with different tense (past vs. present) of the model pre-trained on English (first column) and the adapted models trained on German (second column) and French (third column) at various layers. In all three cases, the projection matrix is computed via PCA on the English representations only.