A Comparative Analysis of Task-Agnostic Distillation Methods for Compressing Transformer Language Models

Takuma Udagawa, Aashka Trivedi, Michele Merler, Bishwaranjan Bhattacharjee
IBM Research AI
{takuma.udagawa@, aashka.trivedi@, mimerler@us., bhatta@us.}ibm.com

Abstract

Large language models have become a vital component in modern NLP, achieving state of the art performance in a variety of tasks. However, they are often inefficient for real-world deployment due to their expensive inference costs. Knowledge distillation is a promising technique to improve their efficiency while retaining most of their effectiveness. In this paper, we reproduce, compare and analyze several representative methods for task-agnostic (general-purpose) distillation of Transformer language models. Our target of study includes Output Distribution (OD) transfer, Hidden State (HS) transfer with various layer mapping strategies, and Multi-Head Attention (MHA) transfer based on MiniLMv2. Through our extensive experiments, we study the effectiveness of each method for various student architectures in both monolingual (English) and multilingual settings. Overall, we show that MHA transfer based on MiniLMv2 is generally the best option for distillation and explain the potential reasons behind its success. Moreover, we show that HS transfer remains as a competitive baseline, especially under a sophisticated layer mapping strategy, while OD transfer consistently lags behind other approaches. Findings from this study helped us deploy efficient yet effective student models for latency-critical applications.

1 Introduction

Large language models have become a crucial component in modern NLP. They have achieved exceptional performance on various downstream tasks (Devlin et al., 2019; Liu et al., 2019; Lewis et al., 2020) and their capability shows consistent improvement with more compute, data, and model parameters (Kaplan et al., 2020; Brown et al., 2020; Touvron et al., 2023). On the downside, it is becoming increasingly difficult to deploy such models in real-world environments due to their inefficiency, i.e. high computation, memory, latency and storage costs (Xu and McAuley, 2023).

Knowledge distillation (Hinton et al., 2015) is a promising technique to overcome this challenge by transferring the knowledge of the original model (teacher) to a smaller, more efficient model (student). This can be conducted in either task-specific (Turc et al., 2019; Jiao et al., 2020) or task-agnostic manner (Sanh et al., 2019; Wang et al., 2020). The latter only requires distilling a single general-purpose student which can be directly finetuned on any downstream task. Due to its high convenience, we focus on this latter approach in this study.

In recent years, there have been various methods proposed for task-agnostic distillation of Transformer language models. The aim of this paper is to reproduce, compare and analyze the most representative methods in this area. We generally focus on the architecture-agnostic distillation which imposes no or minimal restriction on the student architecture: the representative methods include Output Distribution (OD) transfer (Hinton et al., 2015), Hidden State (HS) transfer based on linear mapping (Jiao et al., 2020; Mukherjee et al., 2021) and Multi-Head Attention (MHA) transfer based on MiniLMv2 (Wang et al., 2021).

For HS transfer, the layer mapping strategy between teacher and student layers plays a significant role in overall performance, however, the optimal strategy remains unknown or controversial (Sun et al., 2019; Wu et al., 2020; Ko et al., 2023). Therefore, we explore a diverse range of strategies to empirically evaluate each technique.

For MHA transfer, the MiniLMv2 approach has been shown to achieve state-of-the-art performance, however, there is relatively little understanding behind its success. Therefore, we develop a novel variant named DirectMiniLM which is useful for

\[ \text{By architecture-agnostic, we mean that the student and teacher can have different architectural parameters (e.g. number of layers, attention heads, hidden state size, etc).} \]
understanding the effectiveness behind MiniLMv2 both theoretically and empirically.

In contrast to most previous studies, all methods are reproduced on a single unified codebase for fair and consistent comparison. We also conduct distillation on 4 different student architectures, reducing the model size in various dimensions to fit various parameter and latency budgets. Finally, all experiments are conducted on both monolingual and multilingual settings, distilled from open-source BERT (Devlin et al., 2019) and in-house XLM-RoBERTa (Conneau et al., 2020), respectively.

Through our extensive experiments, we critically analyze the effectiveness of each distillation method and provide practical advice for both researchers and practitioners working in this area. In summary, our key findings are:

- MHA transfer is generally the best option for various student architectures and language settings. By comparison with DirectMiniLM, we provide novel insights underlying its success.
- While the effectiveness of HS transfer depends on the layer mapping strategy, it remains as a competitive baseline. More sophisticated layer mapping strategy can provide a boost in performance, esp. in the multilingual setting.
- Methods relying on OD transfer consistently lag behind other methods. This shows that classical OD distillation can be less effective when distilling complex language models on a general-purpose objective.

## 2 Transformer Language Models

First, we briefly review the standard architecture of Transformer language models (Vaswani et al., 2017; Devlin et al., 2019). A Transformer consists of a stack of $L$ Transformer layers, where each layer comprises two sub-layers: a Multi-Head Attention (MHA) layer followed by a fully connected Feed-Forward (FF) layer (Figure 1, (a)).

Formally, let $x$ denote the input sequence, $d_h$ the hidden state size, and $H_i \in \mathbb{R}^{|x| \times d_h}$ the hidden state of the $i$th Transformer layer ($H_0$ denotes the input sequence embeddings). Given $H_i$, the MHA layer first computes the query, key, and value mappings $Q_{i,a}, K_{i,a}, V_{i,a}$ for each attention head $a \in [1, A_h]$, which are combined to obtain the attention head output $O_{i,a}$:

$$Q_{i,a} = H_i W_{Q,i,a} \quad (1)$$
$$K_{i,a} = H_i W_{K,i,a} \quad (2)$$
$$V_{i,a} = H_i W_{V,i,a} \quad (3)$$
$$O_{i,a} = \text{softmax}(\frac{Q_{i,a} K_{i,a}^T}{\sqrt{d_k}}) V_{i,a} \quad (4)$$

Here, $d_h$ denotes the attention head size (typically set to $\frac{d_f}{A_h}$) and $W_{Q,i,a}, W_{K,i,a}, W_{V,i,a} \in \mathbb{R}^{d_h \times d_h}$ are the learnt weight matrices. The output of the MHA layer is the concatenation of $O_{i,a}$, namely $\text{MHA}(H_i) = \bigoplus_{a=1}^{A_h} O_{i,a}$.

Next, the MHA layer output is followed by a position-wise FF layer with an intermediate size of $d_f$ and a non-linear activation (we use GELU (Hendrycks and Gimpel, 2016) in all models). The hidden state of the next Transformer layer is computed as $H_{i+1} = \text{FF}(\text{MHA}(H_i))$.\(^2\)

Finally, to predict the output distribution over the entire vocabulary $V$, a linear layer $W_O \in \mathbb{R}^{d_f \times |V|}$ is applied on top of the last hidden state to compute the logits $z = H_L W_O \in \mathbb{R}^{|x| \times |V|}$. The output distribution can be obtained by applying the softmax function over $z$, denoted as $\text{softmax}(z)$.

Throughout this paper, we assume that both the student and teacher are Transformer language models with $L^S$ and $L^T$ layers, respectively.

\(^2\)Both MHA and FF sub-layers have a residual connection (He et al., 2016) and are followed by layer normalization (Ba et al., 2016), which are omitted for brevity.
3 Distillation Methods

Next, we introduce the representative task-agnostic distillation methods illustrated in Figure 1, (b-d). For Multi-Head Attention (MHA) transfer, we consider two approaches: MiniLMv2 and its novel variant DirectMiniLM. For a survey of advanced methods and topics we could not cover in this study, please refer to Appendix A.

Output Distribution (OD) Transfer The output distribution of the teacher contains useful information on the relative probabilities of plausible (even if incorrect) predictions (Hinton et al., 2015). In OD transfer, the student is trained to replicate the teacher’s output distribution. This is achieved by optimizing the following loss function, where \( z^S, z^T \) denote the student/teacher logits, CE(\( . \)) the cross entropy loss and \( T \) the output temperature:

\[
L_{OD} = T^2 \cdot CE\left( \text{softmax}\left( \frac{z^T}{T} \right), \text{softmax}\left( \frac{z^S}{T} \right) \right)
\]  

(5)

Hidden State (HS) Transfer Transformer language models progressively learn useful and generalizable features layer by layer. In HS transfer, the student is trained to predict such useful features represented in the teacher’s hidden states.

Formally, each student layer is mapped to a set of teacher layers to be predicted. Let \( \phi(i) \) denote the set mapped from the \( i \)-th student layer, where \( \emptyset \subseteq \phi(i) \subseteq [1, L^T] \). For each \( j \in \phi(i) \), the hidden state of the \( i \)-th student layer \( H^S_i \in \mathbb{R}^{|x| \times d^S_i} \) is linearly transformed to predict the hidden state of the \( j \)-th teacher layer \( H^T_j \in \mathbb{R}^{|x| \times d^T_j} \). This is represented by the following loss function, where \( W^j_i \in \mathbb{R}^{d^S_i \times d^T_j} \) denotes the linear transformation weight and MSE(.) the mean squared error loss:

\[
L_{HS} = \sum_{i=1}^{L^S} \sum_{j \in \phi(i)} \text{MSE}\left( H^S_i W^j_i, H^T_j \right)
\]  

(6)

One open problem in this approach is the choice of layer mapping strategy \( \phi \). We conduct extensive experiments to compare a diverse range of strategies, which will be discussed in §4.

Minilmv2 The MHA layer is a key component in Transformer language models which controls the long-range dependencies and interactions within input texts. Minilmv2 (Wang et al., 2021) is an effective method to deeply transfer this module while allowing different number of attention heads \( A^S_i \) and \( A^T_i \) for the student and teacher. Their main idea is to distil the attention relation matrices (Q-Q, K-K and V-V) obtained by first concatenating the query (Q), key (K), and value (V) mappings from all attention heads and re-splitting them into the same number of attention relation heads \( A_{\alpha} \).

Formally, let \( A^S_{Q,i,a}, A^S_{K,i,a}, A^S_{V,i,a} \in \mathbb{R}^{ |x| \times d^S_i } \) denote the concatenated and re-split queries, keys, and values for the \( j \)-th student layer, where \( a \in [1, A_r] \) and \( d^S_i = \frac{d^S_i}{A_r} \). For instance, \( \bigoplus_{a=1}^{A_r} Q^S_{i,a} = \bigoplus_{a=1}^{A_r} A^S_{Q,i,a} \), i.e. original queries from \( A^S_i \) attention heads are simply concatenated and then re-split into \( A_r \) matrices. We use the same notation for the \( j \)-th teacher layer, \( A^T_{Q,j,a}, A^T_{K,j,a}, A^T_{V,j,a} \in \mathbb{R}^{ |x| \times d^T_j } \), where \( d^T_j = \frac{d^T_j}{A_r} \). Then, the loss function of Minilmv2 can be defined as follows:

\[
L_{MHA} = \sum_{\alpha \in \{ Q,K,V \}} \sum_{a=1}^{A_r} \text{CE}\left( R^T_{\alpha,j,a}, R^S_{\alpha,i,a} \right)
\]  

(7)

\[
R^T_{\alpha,j,a} = \text{softmax}\left( \frac{A^T_{\alpha,i,a} A^T_{Q,K,V,j,a}}{\sqrt{d^T_j}} \right)
\]  

(8)

\[
R^S_{\alpha,i,a} = \text{softmax}\left( \frac{A^S_{\alpha,i,a} A^S_{Q,K,V,i,a}}{\sqrt{d^S_i}} \right)
\]  

(9)

Here, \( R^T_{\alpha,j,a}, R^S_{\alpha,i,a} \in \mathbb{R}^{ |x| \times |x| } \) denote the attention relation matrices which are computed based on the matrix products of \( A^T_{Q,K,V,i,a}, A^S_{Q,K,V,i,a} \) in eq. (8), (9), respectively. Intuitively, this aims to transfer the teacher’s queries (Q), keys (K) and values (V) in a somewhat indirect way through their matrix products (Q-Q, K-K and V-V).

However, there is minimal justification for why this method works effectively. It is also difficult to directly compare the method against HS transfer since the losses are computed differently. To better understand Minilmv2, we propose its novel variant named DirectMiniLM for our analysis.

DirectMiniLM In DirectMiniLM, we aim to transfer the teacher’s Q/K/V mappings more directly through the linear transformation of the student’s ones, just as we did in HS transfer. Specifically, we use the following loss function with the linear transformation \( W_{\alpha,a} \in \mathbb{R}^{d^S_i \times d^T_j} \):

\[
L_{MHA} = \sum_{\alpha \in \{ Q,K,V \}} \sum_{a=1}^{A_r} \text{MSE}\left( A^S_{\alpha,i,a} W_{\alpha,a}, A^T_{\alpha,j,a} \right)
\]  

(10)
DirectMiniLM is important in two aspects. First, this approach is directly comparable to HS transfer based on eq. (6) with the only difference in which information you transfer: the hidden states $H^T_i \rightarrow H^S_j$ or the Q/K/V mappings $A^T_{α,i,a} \rightarrow A^S_{α,j,a}$. From this comparison, we can quantify the precise advantage of transferring each knowledge in an apples-to-apples manner.

Second, DirectMiniLM is also closely relevant to MiniLMv2: if we constrain $W_{α,a}$ to be orthogonal (i.e. $W_{α,a}W^T_{α,a} = I$) and take the matrix product for each term within the MSE loss in eq. (10), we obtain the following loss function:

$$\sum_{α \in \{Q,K,V\}} \sum_{a=1}^{Ar} \text{MSE} \left( A^S_{α,i,a}A^T_{α,i,a}, A^T_{α,j,a}A^T_{α,i,a} \right)$$

This loss closely resembles MiniLMv2 from eq. (7) with a minor difference of using MSE loss instead of CE loss with softmax. Therefore, DirectMiniLM with certain constraints naturally corresponds to MiniLMv2. The major difference is in whether $T^T_{α,i,a}$ is transferred directly (with linear mappings) or indirectly (with relation matrices): by comparing these two approaches, we can precisely quantify the advantage of each optimization technique.

4 Experimental Setup

We explore the task-agnostic knowledge distillation methods under two settings:\footnote{Note that we limit our study to encoder-only models and leave the distillation of decoder-only (Radford et al., 2019) or encoder-decoder (Lewis et al., 2020) models as future work.}

1. Monolingual Distillation: We train English students using the open-source BERT (Devlin et al., 2019) as the teacher. These models are distilled on the same corpus used for pretraining BERT, i.e., English Wikipedia (Devlin et al., 2019) and BookCorpus (Zhu et al., 2015).

2. Multilingual Distillation: We train multilingual students using our in-house XLM-RoBERTa (Conneau et al., 2020) as the teacher, and distill on the CC100 dataset (Conneau et al., 2020), which consists of data in more than 100 languages. We only use a small subset of the corpus to conduct our experiments within a reasonable computation budget while maintaining the language-wise distribution.

In both settings, we use the Base (12 layer) architecture for the teacher, as shown in Table 1. For more details on each distillation setup (e.g. hyperparameters), please refer to Appendix B.

Student Models To conduct a strong comparison of the representative knowledge distillation methods, we train 4 students of varying architectures and latency/parameter budgets. A summary of the student architectures, with their parameters and latency of inference, are shown in Table 1.

Our largest student is a 6 layer model that follows the same architecture as DistilBERT (Sanh et al., 2019). We also use the 6 layer model used in Mukherjee et al. (2021), which has a smaller hidden size than the teacher. Our smaller 4 and 3 layer students were obtained as recommendations from a Neural Architecture Search process (Trivedi et al., 2023) to find good student architectures for distillation from the XLM-RoBERTa teacher, conditioned to minimize the latency on CPU. Please refer to Appendix C for more details.

Layer Mapping Strategies The layer mapping strategy $φ$ is a parameter that needs to be considered for both HS and MHA transfer. For HS transfer, we explore the following three settings:

1. Single Mapping: We only distil the last ($L^T_{th}$) teacher layer into the last student layer, which has been shown to be a simple yet competitive baseline (Ko et al., 2023).

2. 1-to-1 Mapping: Prior work shows that mapping not only the last layer but also the intermediate layers improves distillation (Sun et al., 2019). In 1-to-1 mapping, we distil one teacher layer into each student layer by choosing:
   - **Last** $L^S$ teacher layers, i.e. $φ(i) = \{L^T - L^S + i\}$ ($i \in [1, L^S]$). Empirically, last teacher layers capture more high-level (e.g. semantic) knowledge in their representations (Tenney et al., 2019; Jawahar et al., 2019).
   - A **Uniform** selection of teacher layers which chooses every $k^{th}$ teacher layer, i.e. $φ(i) = \{ki\}$, where $k = \lceil L^T / L^S \rceil$.\footnote{This strategy is used in DistilBERT (Sanh et al., 2019) and also known as the ”skip” strategy (Sun et al., 2019).} This method can also transfer the lower teacher layers, which empirically captures local (e.g. syntactic) knowledge (Tenney et al., 2019).

3. 1-to-N Mapping: Some works even show that mapping each student layer to multiple teacher layers can avoid the loss of information and facilitate student learning (Wu et al., 2020; Passban et al., 2021). For 1-to-N Mapping, we ex-
We include the results of CoLA in Appendix D. Specifically, we experiment with the last teacher layer for MiniLMv2 and DirectMiniLM.

Table 2 summarizes the performance of each distillation method on 4 student architectures. For detailed evaluations of each method based on the performance on 15 languages after being finetuned on only English training data. We report the average score of all languages for XNLI.

Table 3 summarizes the performance of each distillation method for 4 student architectures. For detailed evaluations of each method based on the configuration, please refer to Appendix D. We also provide a comparison against DistilBERT (Sanh et al., 2019), a representative architecture-constrained method, in Appendix E.

**HS Transfer** From Table 3, we can verify that the performance of HS transfer varies with different layer mapping strategies, and no strategy dominates the others in all settings. In the monolingual setting, we found that the single mapping strategy performs competitively, which is in line with the findings of Ko et al. (2023). However, in the multilingual setting, more sophisticated 1-to-N strategies generally show superiority over the simpler baselines. This indicates that more supervision from the teacher can be helpful (and at worst harmless), hence we advocate for the adoption 1-to-N strategies, esp. in the challenging multilingual distillation.

**OD Transfer** As mentioned in §4, we initialize the model from the HS transferred checkpoints with each layer mapping strategy. Interestingly, we see a slight degradation in performance on downstream tasks compared to only HS transfer, with a signifi-

---

6Our 6L monolingual student takes 49 hours on 30 V100 GPUs to reach acceptable performance, while the same model achieves better scores in only 10.5 hours when initialized from the HS transferred checkpoint.

7Distilled models often perform poorly on CoLA: We hypothesize this is because CoLA is the only syntactic task in the benchmark as opposed to the other semantic tasks (Xu et al., 2022). We include the results of CoLA in Appendix D.
### Table 3: Performance of the representative distillation methods evaluated on avg. GLUE and XNLI. Results based on the best layer mapping strategy for each method is underlined, and the best overall result is shown in bold.

<table>
<thead>
<tr>
<th>Distillation Method</th>
<th>Layer Mapping Strategy</th>
<th>Avg. GLUE (Monolingual)</th>
<th>Avg. GLUE (Multilingual)</th>
<th>Avg. XNLI (Multilingual)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>6L- DistilBERT</td>
<td>6L</td>
<td>4L</td>
</tr>
<tr>
<td>HS Transfer</td>
<td>( L^T )</td>
<td>84.1</td>
<td>79.4</td>
<td>80.2</td>
</tr>
<tr>
<td></td>
<td>Last</td>
<td>83.2</td>
<td>80.4</td>
<td>79.3</td>
</tr>
<tr>
<td></td>
<td>Uniform</td>
<td>82.9</td>
<td>80.6</td>
<td>79.6</td>
</tr>
<tr>
<td></td>
<td>Uniform-Cons.</td>
<td>83.9</td>
<td>80.6</td>
<td>80.6</td>
</tr>
<tr>
<td></td>
<td>Uniform+Last</td>
<td>84.1</td>
<td>80.4</td>
<td>80.4</td>
</tr>
<tr>
<td>OD Transfer (init. from HS Transfer)</td>
<td>( L^T )</td>
<td>84.1</td>
<td>78.1</td>
<td>79.4</td>
</tr>
<tr>
<td></td>
<td>Last</td>
<td>83.1</td>
<td>80.4</td>
<td>79.3</td>
</tr>
<tr>
<td></td>
<td>Uniform</td>
<td>83.4</td>
<td>79.8</td>
<td>79.8</td>
</tr>
<tr>
<td></td>
<td>Uniform-Cons.</td>
<td>83.7</td>
<td>80.3</td>
<td>79.5</td>
</tr>
<tr>
<td></td>
<td>Uniform+Last</td>
<td>84.1</td>
<td>80.5</td>
<td>79.9</td>
</tr>
<tr>
<td>MiniLMv2</td>
<td>( L^T )</td>
<td>84.2</td>
<td>81.9</td>
<td>79.9</td>
</tr>
<tr>
<td></td>
<td>( (L^T -1)^{th} ) ( L^T )</td>
<td>84.2</td>
<td>82.5</td>
<td>80.0</td>
</tr>
<tr>
<td></td>
<td>( (L^T -2)^{th} ) ( L^T )</td>
<td>84.4</td>
<td>82.2</td>
<td>\textbf{80.7}</td>
</tr>
<tr>
<td>DirectMiniLM</td>
<td>( L^T )</td>
<td>84.0</td>
<td>81.3</td>
<td>79.7</td>
</tr>
<tr>
<td></td>
<td>( (L^T -1)^{th} ) ( L^T )</td>
<td>\textbf{84.4}</td>
<td>81.7</td>
<td>79.6</td>
</tr>
<tr>
<td></td>
<td>( (L^T -2)^{th} ) ( L^T )</td>
<td>84.3</td>
<td>81.7</td>
<td>80.4</td>
</tr>
<tr>
<td>Teacher</td>
<td></td>
<td>85.5</td>
<td>84.8</td>
<td>84.8</td>
</tr>
</tbody>
</table>

**MHA Transfer**

For both MiniLMv2 and DirectMiniLM, we found distilling the upper-middle teacher layer, i.e. \( (L^T -1)^{th} \) or \( (L^T -2)^{th} \) strategy, led to the best performance, in line with the original findings of Wang et al. (2021). Importantly, we found that both MHA transfer methods generally outperform HS transfer, which points to the benefit of transferring the Q/K/V knowledge over the hidden state knowledge. This is consistent with the latest comparative study by Wang et al. (2023), although they only evaluate on the 6L-DistilBERT architecture in the monolingual setting.

We also note that MiniLMv2 and DirectMiniLM perform equivalently, with the notable exception on XNLI. We attribute this to two factors:

1. MiniLMv2 transfers relational representations conditioned on the whole input, while DirectMiniLM transfers absolute position-wise representations. The former may be more semantically informative, as the contextual representations often exhibit rich relational structures (Park et al., 2021; Liu et al., 2022a).
2. DirectMiniLM requires learning the linear transformation weight \( W_{\alpha, \beta} \), while MiniLMv2 does not incur any additional parameters.

From these observations, we generally expect MiniLMv2 to be the best distillation method and have adopted it in our latency-critical applications.\(^8\) However, DirectMiniLM performs comparably and provides meaningful insights on the benefit of each optimization technique, which can be useful for debugging and analyzing MiniLMv2. Therefore, we recommend its comparison for both researchers and practitioners in future studies.

### 6 Conclusion

This study critically analyzes the representative methods for task-agnostic distillation of language models. Specifically, we compare Output Distribution (OD), Hidden State (HS), and Multi-Head Attention (MHA) transfer for different student architectures, language settings, and layer mapping strategies. Through our extensive experiments, we show that MHA transfer based on MiniLMv2 is the best option across many settings, followed by HS transfer with sophisticated 1-to-N mapping strategies. Meanwhile, we did not find OD transfer to be an effective alternative. Finally, we propose DirectMiniLM to demystify the precise advantage of the indirect (i.e. relation matrix based) optimization technique proposed in MiniLMv2. Overall, we hope this study will be a useful guide for both researchers and practitioners working in this area.

\(^8\)Specifically, the 4L monolingual and multilingual students with 7x speedup on CPU have been deployed for various NLP applications, such as entity extraction, document classification and relation detection, while maintaining 93% of the teacher’s performance on average (Trivedi et al., 2023).
References


A Related Work

MobileBERT (Sun et al., 2020) is an effective technique to compress BERT into a specially designed student with a bottleneck architecture. In BERT-of-Theseus (Xu et al., 2020), the modules of the teacher are progressively replaced with smaller ones to improve efficiency. However, these approaches constrain the architecture of the students. In contrast, we focus on the architecture-agnostic distillation methods for better flexibility.

Improvements on distillation objectives are also made, e.g. transferring the relational, structural or holistic representations of the language models may provide more useful signals for students (Park et al., 2021; Liu et al., 2022a; Tan et al., 2023). When the transfer set is limited, various methods of data augmentation (Liang et al., 2021; Zhang et al., 2022; Liu et al., 2022b) can be applied successfully. To alleviate the capacity gap between the teacher and student, previous works proposed scheduled annealing in OD transfer (Jafari et al., 2021), multi-stage distillation with intermediate-sized teacher assistants (Mirzadeh et al., 2020; Son et al., 2021), and meta-learning to optimize the teacher for student distillation (Zhou et al., 2022; Ma et al., 2022). We leave the exploration of such advanced techniques as future work.

Layer mapping strategies for HS transfer have also been studied extensively. Jiao et al. (2021) proposed an evolutionary search process to obtain the optimal layer mapping for specific downstream tasks. Li et al. (2020) applied Earth Mover’s Distance to prioritize mappings with smaller cost (i.e. distillation loss). The attention mechanism can also be applied to map student layers to similar teacher layers, where the similarity is computed based on the cosine similarity (Passban et al., 2021) or the predictions of internal classifiers (Wu et al., 2021). Finally, random mapping has been shown to work surprisingly well, potentially working as a regularizer to prevent overfitting (Haidar et al., 2022).
In this study, we focus instead on the carefully designed and easily applicable heuristic strategies. Finally, there are different approaches to reducing the inference costs of large language models, such as quantization (Zafir et al., 2019; Shen et al., 2020; Kim et al., 2021; Bai et al., 2021), pruning (Fan et al., 2020; Laguna et al., 2021; Xia et al., 2022), early exit mechanisms (Liu et al., 2020; Xin et al., 2021; Liao et al., 2021; Wang et al., 2022), and matrix decomposition (Ben Noach and Goldberg, 2020; Mao et al., 2020; Chen et al., 2021; Tahaei et al., 2022). Many of these approaches are complementary to our distillation methods and can be combined for further efficiency.

B Distillation Setup

We train our monolingual students on the entire Wikipedia and BookCorpus using the AdamW Optimizer (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9, \beta_2 = 0.98$. For HS and MHA transfer, students are trained for 7 epoch with a peak learning rate (LR) of $5e - 4$. For OD transfer, we train for 3 epochs with a peak LR of $3e - 4$ after HS transfer. We use a linear LR warmup over the first 5% of the training steps and then a linear decay. We use a batch size of 32 with the maximum sequence length set to 256 and train on 30 V100 GPUs.

For multilingual distillation, we use a small subset of CC-100 containing 7M sentences, which we found to be sufficient for developing competitive students. We generally use the same setup as monolingual distillation, except we use the peak LR of $8e - 4$ for MHA transfer. Multilingual students are trained on 2 A100-80GB GPUs.

Finally, the method-specific hyperparameters (§3) are as follows. For OD transfer, we set the output temperature $T$ to the default value of 1. For MiniLMv2, we use $A_r > A_h$ to transfer more fine-grained knowledge in the Q/K/V mappings: specifically, we set $A_r = 48$, which is also used in Wang et al. (2021). For DirectMiniLM, we found using $A_r = A_h$ without the orthogonal constraints on $W_{a,a}$ led to the best performance and used this setting throughout our experiments.

C Finding Smaller Student Models

Our smallest students, a 4 layer and a 3 layer model, were obtained as recommendations from a Neural Architecture Search process to find good student architectures for task-agnostic distillation from an XLM-RoBERTa teacher, conditioned to minimize the latency of inference on a CPU. Specifically, we follow the KD-NAS method of Trivedi et al. (2023) and modify the reward to reduce the distillation loss $L_{HS}$ defined in Eq. (6), along with the CPU latency of the student $(lat(S))$ normalized by the teacher’s latency $(lat(T))$:

$$reward(S) = (1 - L_{HS}) \times \left( \frac{\text{lat}(S)}{0.6 + \text{lat}(T)} \right)^{-0.06}$$

Please refer to their original paper for more details.

D Evaluation Results for Best Models

We include detailed results of each distillation method for the best configuration (i.e. layer mapping strategy). Specifically, we show the results of each GLUE task for monolingual and multilingual distillation in Table 5 and 6. We show language-wise performance on XNLI in Table 7. All downstream tasks are evaluated on 3 random seeds.

For the sake of efficient evaluation, we did not conduct expensive grid search for finetuning hyperparameters. After some manual tuning, we used the same LR of $2e - 5$ and batch size of 32 for finetuning all models on all tasks. We used 3 epochs of finetuning for GLUE tasks (except CoLA, where we used 6 and 10 epochs for monolingual and multilingual models) and 5 epochs for XNLI.

E Architecture Constrained Distillation: DistilBERT

DistilBERT (Sanh et al., 2019) is one of the earliest and most widely used baseline. This method comprises (1) layer initialization from the teacher layers, (2) HS transfer based on cosine similarity loss, and (3) OD transfer. The first two techniques restrict the architecture of each student layer to be identical to the teacher model, which limits our analysis to the 6L-DistilBERT student architecture.

<table>
<thead>
<tr>
<th>6L-DistilBERT</th>
<th>Teacher</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. GLUE (Monolingual)</td>
<td>82.9 (0.5)</td>
</tr>
<tr>
<td>Avg. GLUE (Multilingual)</td>
<td>79.7 (0.5)</td>
</tr>
<tr>
<td>Avg. XNLI (Multilingual)</td>
<td>61.8 (0.5)</td>
</tr>
</tbody>
</table>

Table 4: DistilBERT Performance. Average GLUE scores reported for all tasks w/o CoLA. Average XNLI scores reported for all languages. Average taken over 3 random seeds with standard deviation in parenthesis.

As shown in the results of Table 4, the performance of DistilBERT is generally not competitive with our distillation methods from Table 3, especially in the multilingual setting.
Table 5: Monolingual Student GLUE Performance for all tasks. Each row shows performance based on the best layer mapping strategy. Each score reported as an average over 3 random seeds (standard deviation in parenthesis).

<table>
<thead>
<tr>
<th>Model</th>
<th>Distillation Method</th>
<th>Best Strategy</th>
<th>MNLI</th>
<th>QQP</th>
<th>QNLI</th>
<th>SST-2</th>
<th>CoLA</th>
<th>STS-B</th>
<th>MRPC</th>
<th>RTE</th>
<th>Avg.</th>
<th>Avg. (-CoLA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6L-DistilBERT</td>
<td>HS Transfer</td>
<td>Uniform-Last</td>
<td>82.6</td>
<td>86.2</td>
<td>88.7</td>
<td>90.8</td>
<td>45.9</td>
<td>85.9</td>
<td>67.7</td>
<td>63.5</td>
<td>62.5</td>
<td>79.4 (0.5)</td>
</tr>
<tr>
<td></td>
<td>OD Transfer</td>
<td>Uniform-Last</td>
<td>82.7</td>
<td>86.5</td>
<td>88.3</td>
<td>91.3</td>
<td>50.8</td>
<td>85.5</td>
<td>69.7</td>
<td>64.2</td>
<td>64.2</td>
<td>79.9 (0.3)</td>
</tr>
<tr>
<td></td>
<td>DirectMiniLM</td>
<td>(L&lt;sub&gt;T-1&lt;/sub&gt;)&lt;sup&gt;TH&lt;/sup&gt;</td>
<td>82.9</td>
<td>86.6</td>
<td>90.0</td>
<td>91.4</td>
<td>52.7</td>
<td>86.4</td>
<td>69.0</td>
<td>64.7</td>
<td>64.7</td>
<td>80.5 (0.4)</td>
</tr>
<tr>
<td></td>
<td>HS Transfer</td>
<td>Uniform-Cons.</td>
<td>79.8</td>
<td>85.0</td>
<td>85.9</td>
<td>90.9</td>
<td>31.2</td>
<td>83.2</td>
<td>64.4</td>
<td>56.3</td>
<td>56.3</td>
<td>74.4 (0.4)</td>
</tr>
<tr>
<td></td>
<td>OD Transfer</td>
<td>Uniform-Cons.</td>
<td>79.1</td>
<td>84.6</td>
<td>86.3</td>
<td>89.7</td>
<td>38.6</td>
<td>82.3</td>
<td>63.7</td>
<td>57.9</td>
<td>57.9</td>
<td>75.3 (0.6)</td>
</tr>
<tr>
<td></td>
<td>DirectMiniLM</td>
<td>(L&lt;sub&gt;T-1&lt;/sub&gt;)&lt;sup&gt;TH&lt;/sup&gt;</td>
<td>80.8</td>
<td>84.9</td>
<td>88.0</td>
<td>90.3</td>
<td>36.2</td>
<td>84.5</td>
<td>68.2</td>
<td>62.5</td>
<td>62.5</td>
<td>76.7 (0.1)</td>
</tr>
<tr>
<td></td>
<td>HS Transfer</td>
<td>Uniform-Cons.</td>
<td>78.2</td>
<td>84.6</td>
<td>85.1</td>
<td>90.1</td>
<td>32.2</td>
<td>83.3</td>
<td>65.1</td>
<td>55.1</td>
<td>55.1</td>
<td>74.0 (0.2)</td>
</tr>
<tr>
<td></td>
<td>OD Transfer</td>
<td>Uniform-Cons.</td>
<td>78.8</td>
<td>83.8</td>
<td>86.0</td>
<td>90.8</td>
<td>30.9</td>
<td>83.0</td>
<td>64.3</td>
<td>58.2</td>
<td>58.2</td>
<td>74.5 (0.2)</td>
</tr>
</tbody>
</table>

Table 6: Multilingual Student GLUE Performance for all tasks. Each row shows performance based on the best layer mapping strategy. Each score reported as an average over 3 random seeds (standard deviation in parenthesis).

<table>
<thead>
<tr>
<th>Model</th>
<th>Distillation Method</th>
<th>Best Strategy</th>
<th>ar</th>
<th>bg</th>
<th>de</th>
<th>el</th>
<th>en</th>
<th>es</th>
<th>fr</th>
<th>hi</th>
<th>hu</th>
<th>sv</th>
<th>th</th>
<th>tr</th>
<th>ur</th>
<th>vi</th>
<th>zh</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>6L-DistilBERT</td>
<td>HS Transfer</td>
<td>Uniform-Last</td>
<td>64.7</td>
<td>69.7</td>
<td>69.6</td>
<td>69.2</td>
<td>80.7</td>
<td>72.0</td>
<td>70.2</td>
<td>64.2</td>
<td>67.7</td>
<td>51.2</td>
<td>65.3</td>
<td>62.5</td>
<td>58.9</td>
<td>70.4</td>
<td>68.6</td>
<td>67.0 (0.4)</td>
</tr>
<tr>
<td></td>
<td>OD Transfer</td>
<td>Uniform-Last</td>
<td>63.7</td>
<td>69.4</td>
<td>67.0</td>
<td>78.6</td>
<td>70.7</td>
<td>68.9</td>
<td>60.0</td>
<td>68.9</td>
<td>60.2</td>
<td>51.2</td>
<td>65.3</td>
<td>61.9</td>
<td>57.9</td>
<td>68.5</td>
<td>68.8</td>
<td>66.0 (0.6)</td>
</tr>
<tr>
<td></td>
<td>DirectMiniLM</td>
<td>(L&lt;sub&gt;T-1&lt;/sub&gt;)&lt;sup&gt;TH&lt;/sup&gt;</td>
<td>65.5</td>
<td>71.6</td>
<td>72.1</td>
<td>71.5</td>
<td>84.1</td>
<td>75.0</td>
<td>73.5</td>
<td>65.3</td>
<td>70.6</td>
<td>65.1</td>
<td>67.1</td>
<td>61.0</td>
<td>69.7</td>
<td>69.7</td>
<td>69.3</td>
<td>69.1 (0.5)</td>
</tr>
<tr>
<td></td>
<td>HS Transfer</td>
<td>Uniform-Cons.</td>
<td>70.9</td>
<td>87.2</td>
<td>83.4</td>
<td>85.8</td>
<td>79.9</td>
<td>66.8</td>
<td>58.3</td>
<td>62.4</td>
<td>48.9</td>
<td>62.7</td>
<td>59.1</td>
<td>59.6</td>
<td>63.4</td>
<td>63.2</td>
<td>65.1</td>
<td>62.7 (0.4)</td>
</tr>
<tr>
<td></td>
<td>OD Transfer</td>
<td>Uniform-Cons.</td>
<td>75.7</td>
<td>86.0</td>
<td>83.2</td>
<td>84.5</td>
<td>8.7</td>
<td>70.6</td>
<td>61.7</td>
<td>87.1</td>
<td>57.3</td>
<td>62.8</td>
<td>62.2</td>
<td>78.3 (0.5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DirectMiniLM</td>
<td>(L&lt;sub&gt;T-1&lt;/sub&gt;)&lt;sup&gt;TH&lt;/sup&gt;</td>
<td>72.2</td>
<td>81.2</td>
<td>83.4</td>
<td>84.8</td>
<td>15.9</td>
<td>67.9</td>
<td>62.0</td>
<td>58.0</td>
<td>58.0</td>
<td>68.2 (1.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Multilingual Student XNLI Performance for 15 languages. Each row shows performance based on the best layer mapping strategy. Each score reported as an average over 3 random seeds (standard deviation in parenthesis).