Align-to-Distill: Trainable Attention Alignment for Knowledge Distillation in Neural Machine Translation

**Heegon Jin, Seonil Son, Jemin Park, Youngseok Kim,
Hyungjong Noh, Yeonsoo Lee**

NC Research, AI Tech Center
Seongnam, South Korea
{heegon, deftson, hbk5844, kys159, nohhj0209, yeonsoo}@ncsoft.com

Abstract

The advent of scalable deep models and large datasets has improved the performance of Neural Machine Translation (NMT). Knowledge Distillation (KD) enhances efficiency by transferring knowledge from a teacher model to a more compact student model. However, KD approaches to Transformer architecture often rely on heuristics, particularly when deciding which teacher layers to distill from. In this paper, we introduce the "Align-to-Distill" (A2D) strategy, designed to address the feature mapping problem by adaptively aligning student attention heads with their teacher counterparts during training. The Attention Alignment Module (AAM) in A2D performs a dense head-by-head comparison between student and teacher attention heads across layers, turning the combinatorial mapping heuristics into a learning problem. Our experiments show the efficacy of A2D, demonstrating gains of up to +3.61 and +0.63 BLEU points for WMT-2022 De→Dsb and WMT-2014 En→De, respectively, compared to Transformer baselines. The code and data are available at `https://github.com/ncsoft/Align-to-Distill`.

1. Introduction

Transformer-based encoder-decoder models have achieved remarkable success in various natural language processing tasks (Vaswani et al., 2017; Devlin et al., 2019; Liu et al., 2019; Lewis et al., 2020), including Neural Machine Translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015; Sennrich et al., 2016; Vaswani et al., 2017). However, the autoregressive decoding process often imposes a significant computational burden, especially as the number of layers and parameters escalates with increasing model complexity. This presents substantial challenges when deploying the Transformer-based models for real-time applications (Gu et al., 2017) and online services (Zhou et al., 2022).

One of the possible solutions is to reduce the model size using knowledge distillation (KD) (Hinton et al., 2015). KD facilitates the transfer of knowledge from a high-performing, large-parameter teacher model to a more moderately sized student model. This process alleviates deployment challenges by generating a distilled model that is both lightweight and efficient, ensuring reduced inference times and lower computational resource requirements. Furthermore, with the guidance of the teacher model, the student model can potentially achieve performance levels closer to those of the teacher model compared to training it without the teacher's assistance.

KD, initially proposed by Bucila et al.; Ba and Caruana; Hinton et al., involves transferring knowledge to the student model using responses from the network's last layer. Among its variants, Sequence-level KD (Kim and Rush, 2016) and Selective KD (Wang et al., 2021) leverage the final output and soft labels from teacher's responses, respectively. These strategies can be categorized as response-based KD (Gou et al., 2021). Meanwhile, not only the final layer outputs are used, but intermediate features from the teacher model's layer are also used as a medium for a more effective and comprehensive distillation of knowledge (Romero et al., 2015; Zagoruyko and Komodakis, 2017; Sun et al., 2019; Jiao et al., 2020; Sun et al., 2020). These approaches belong to the category of feature-based KD (Gou et al., 2021). Most feature-based KD in Transformers has concentrated on compressing encoder-based models (Sanh et al., 2019; Sun et al., 2019; Jiao et al., 2020; Wang et al., 2020; Sun et al., 2020; Passban et al., 2021), including pre-trained models like BERT (Devlin et al., 2019). On the other hand, some studies (Wu et al., 2020; Shleifer and Rush, 2020) have applied feature-based KD to the decoder for generative tasks. However, they found it less effective compared to response-based KD for decoder distillation (Kim and Rush, 2016; Kasai et al., 2020; Wang et al., 2021).

While extending KD to features across the layers does enrich knowledge transfer, it prompts an open question: 'From which teacher layer should the student layer learn and from which should it not?'. Instead of resolving this issue through trainable methods, several studies (Sun et al., 2019; Jiao et al., 2020; Wu et al., 2020; Passban et al.,

2021) have circumvented the issue using heuristic approaches. Those approaches require a heuristic skip or combination of teacher layers to align with the student layer. However, as the number of layers increases, the complexity of heuristically selecting features grows, necessitating an exhaustive search for the optimal combination strategy. For example, Combinatorial KD (Wu et al., 2020) demonstrated that its peak performance relies on language-pair-specific feature mapping.

In this paper, we introduce a novel KD strategy, Align-to-Distill (A2D), that addresses the feature mapping problem using a trainable Attention Alignment Module (AAM). Unlike earlier KD methods that relied on combinatorial feature mapping heuristics, A2D provides an end-to-end trainable solution. The adaptive alignment of features removes the necessity for a data-dependent mapping strategy. Furthermore, AAM aligns the student attention map in each head with those of the teacher, resulting in more effective distillation compared to layer-wise feature mapping. AAM enables each attention head in the student model to be compared with every head in the teacher model across different layers, by employing pointwise convolution with only a few additional parameters. As a result, there is no longer a need for head or layer parity between the student and teacher models.

Notably, our experimental results and analysis show that due to its fine-grained attention transfer in a head-wise manner, A2D is effectively applicable to the decoder of the Transformer, an area where previous feature-based KD approaches have typically struggled. By compressing the decoder with A2D, we could reduce the cost of autoregressive inference while preserving its performance. Our comprehensive studies on both high-resource and low-resource NMT tasks show that our method consistently outperforms state-of-the-art baselines, spanning both feature-based and response-based KD methods. In particular, even with a smaller model size than the teacher, students trained with A2D can match or even surpass teacher performance in low-resource settings.

Our contribution is three-folded as follows:

- We introduce "Align-to-Distill" (A2D), a novel attention-based distillation method that can be effectively applied to the decoder of the transformer.

- A2D overcomes the limitations imposed by feature-mapping heuristics of previous distillation approaches by introducing a learnable alignment between attention heads across different layers.

- A2D enables fine-grained attention knowledge transfer from teacher to student, thereby outperforming state-of-the-art KD strategies in both high-resource and low-resource translation tasks.

2. Preliminaries

2.1. Multi-Head Attention

Multi-head Attention (MHA) is a core component of the Transformer architecture introduced by Vaswani et al.. We use $\mathbf{X} \in \mathcal{R}^{L \times d}$ to denote the input to the Transformer layer, which is a sequence of embeddings fed into MHA, where L is the length of the text input and d is the dimension of model embedding.

In the Transformer layer, the input sequence $\mathbf{X}$ is mapped to three unique vector embeddings: the query ($\mathbf{Q}$), the key ($\mathbf{K}$), and the value ($\mathbf{V}$). Each of these embeddings is associated with its own projection weight, denoted by $\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v$ respectively.

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_q, \mathbf{K} = \mathbf{X}\mathbf{W}_k, \mathbf{V} = \mathbf{X}\mathbf{W}_v \qquad (1)$$

Note that these projection weights ($\mathbf{W}_{q|k|v}$) are unique for each attention head and layer. The shape of $\mathbf{W}_{q|k|v}$ is $\mathcal{R}^{d \times d_{\text{head}}}$, which results in $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V} \in \mathcal{R}^{L \times d_{\text{head}}}$. Head dimension d_{head} is d divided by the number of attention heads in a layer.

Scaled dot-product attention, $\text{attn}(\mathbf{X})$, that each MHA head computes is defined as follows:

$$\text{attn}(\mathbf{X}) = s\left(\mathbf{Q}\mathbf{K}^\top / \sqrt{d_{\text{head}}}\right)\mathbf{V} = \mathbf{H}\mathbf{V} \in \mathcal{R}^{L \times d_{\text{head}}}$$
$$(2)$$

where $s(\cdot)$ denotes a softmax function. The attention output denoted by $\text{attn}(\mathbf{X})$ is concatenated over d_{head} to form MHA of shape $\mathcal{R}^{L \times d}$.

Attention map denoted by $\mathbf{H}$ above, is the scaled dot-product of the query-key for each head, represented as $s(\mathbf{Q}\mathbf{K}^\top / \sqrt{d_{\text{head}}}) \in \mathcal{R}^{L \times L}$. To make clearer notion of distinct heads and layers, throughout the paper the attention map will appear as:

$$\mathbf{H}_{(m,n)} = s\left(\mathbf{Q}_{(m,n)}\mathbf{K}_{(m,n)}^\top / \sqrt{d_{\text{head}}}\right) \in \mathcal{R}^{L \times L} \qquad (3)$$

where m and n serve as indices for head and layer inside the whole Transformer architecture. Thus, $\mathbf{H}_{(m,n)}$ represents as attention map m-th head in n-th Transformer layer. Similarly $\mathbf{Q}_{(m,n)}$, $\mathbf{K}_{(m,n)}$, and $\mathbf{V}_{(m,n)}$ refer to the query, key, and value of the same head.

2.2. Knowledge Distillation

The idea of knowledge distillation from a larger neural network to a smaller network was proposed in (Hinton et al., 2015). In the paper, the output

distribution of the larger teacher network is used as a soft target for training the student network.

$$\mathcal{L}_{\text{KD}} = -\sum_{x \in \mathcal{X}} p^T \log(p^S) \tag{4}$$

where p^T and p^S denote the output distributions of the teacher and student. The core idea of training the student to imitate the teacher extends further from comparing the output distribution to any valuable intermediate features. In general, the following loss function provides an abstraction.

$$\mathcal{L} = \sum_{x \in \mathcal{X}} \text{D}(f^T(x), f^S(x)) \tag{5}$$

Equation (5) compares the features of the teacher and student models ($f^T(x)$, $f^S(x)$) for a given input x from a dataset $\mathcal{X}$ with a measure of dissimilarity (D). There are a number of widely used choices for features to distill knowledge from, such as hidden states (Sun et al., 2019; Passban et al., 2021) or attention maps (Jiao et al., 2020; Wu et al., 2020). A common choice for D is Kullback-Leibler Divergence (Joyce, 2011) or mean-squared error. As the student network learns to imitate the features from the teacher with Equation (5), the teacher model's knowledge, represented by the feature $f^T(x)$, is transferred to the student. This transfer of knowledge helps the student model converge to a better optimum that it could not reach on its own.

3. Related Works

In the field of feature-based KD studies, which incorporate intermediate representations of the teacher for KD, an aligning strategy has been explored to handle different architectural settings between the teacher and student. Since the teacher model generally has more layers than the student model, Patient KD (Sun et al., 2019) proposed skipping every other layer or using only the last consecutive layers of the teacher model so that the number of matching layers becomes equal. Likewise, TinyBERT (Jiao et al., 2020) formulated a layer-mapping function that selects hint layers to match the student layers at a similar depth level. Meanwhile, MiniLM (Wang et al., 2020) circumvented the aligning issue by distilling only the last layer. However, it constrained the number of heads to be equivalent between teacher and student networks.

To address the issue of skipped teacher layers, CKD (Wu et al., 2020) proposed layer fusion, a method that projects several teacher layers into one fused layer. The fusion is then matched with the corresponding student layer, enabling distillation from all teacher layers. Although CKD benefits from the representations contained in more teacher layers, the mapping between fused teacher layers

and student layers is set heuristically, before training. As a result, its optimal combination varies due to the training dataset, and the number of cases for mapping becomes intractable when the number of layers increases. Additionally, performance degradation was reported when the method was applied to the decoder part of the Transformer, regardless of the combination setting. This led to CKD being applied solely to the encoder side of the Transformer. While ALP-KD (Passban et al., 2021) made the mapping partially adaptive with an attention mechanism, it still requires dividing student layers into buckets, which involves heuristics, and the result is dependent on how the buckets are grouped. Both CKD and ALP-KD are layer-wise fusion distillation methods and do not consider more detailed mappings which can be found in attention heads.

4. Methodology

In this section, we provide a detailed overview of the architecture and training of the Attention Alignment Module (AAM), the core component of our Align-to-Distill (A2D) approach. Figure 1 illustrates the overall framework.

4.1. Attention Map as a Knowledge

The attention mechanism is a vital component in Transformer models, capturing the context and relationships between words in a sentence. Attention alignment ensures that the student model, despite potential differences in network structure compared to the teacher model, focuses on the same word relationships and contextual nuances as the teacher. This process allows the student model to gain insights beyond mere output mimicry; it learns the underlying relationships that the teacher model uses for prediction, effectively transferring knowledge from the teacher model to the student.

A2D compares the attention maps in Equation (3) from the teacher and student for knowledge distillation. As depicted previously the shape of the attention map is determined by the length of the input (i.e. $\mathbf{H}_{(m,n)} \in \mathcal{R}^{L \times L}$), not by the model hyperparameters. This eases the comparison between knowledge features between the teacher and student models on a different scale.

In the encoder-decoder model, there exist three distinct attention that computes relation within, and in between the encoder and decoder input sequences: self-attention within the encoder and decoder, and cross-attention from encoder to decoder. The A2D method incorporates all three types of at-
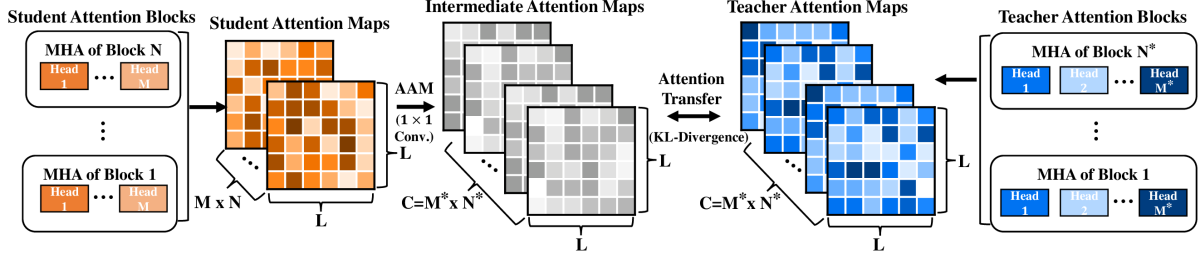
Figure 1: Attention Transfer with A2D. The Attention Alignment Module (AAM), implemented as a pointwise convolution layer, produces intermediate attention maps from a collection of student attention maps. The number of intermediate maps matches the total attention maps of the teacher model, encompassing all layers and heads. These intermediate maps are then directly compared to the teacher's attention maps using KL-Divergence, without any form of reduction.

tention maps for effective knowledge distillation:

$$H^{\text{enc}-\text{self}}_{(m,n)} \in \mathcal{R}^{L_{\text{src}} \times L_{\text{src}}}, \tag{6}$$

$$H^{\text{dec}-\text{self}}_{(m,n)} \in \mathcal{R}^{L_{\text{tgt}} \times L_{\text{tgt}}}, \tag{7}$$

$$H^{\text{dec}-\text{cross}}_{(m,n)} \in \mathcal{R}^{L_{\text{src}} \times L_{\text{tgt}}} \tag{8}$$

Note that L_{src} and L_{tgt} are the lengths of the source and target sentences for machine translation, which make the shape of attention maps among three types of attention. During the attention knowledge transfer, each type of attention map is compared between teacher and student, respectively.

4.2. Attention Alignment Module

4.2.1. Module Architecture

The student and teacher models share the same L_{src} and L_{tgt} during distillation, leading to attention maps of the same shape for both models, regardless of model differences. However, the number of attention maps in these models differs since the teacher model has more layers and more heads in each layer compared to the student model. This discrepancy in numbers makes it infeasible to make a straightforward one-to-one mapping of attention maps between teacher and student models. To resolve this, we devise an Attention Alignment Module (AAM) that generates intermediate attention maps, bridging the gap between the two groups of attention maps.

AAM performs pointwise convolution (1×1 Conv. as in Figure 1) on the student attention maps. This operation creates an equal number of intermediate attention maps to the teacher maps. Each intermediate map results from a weighted sum of all student attention maps. Consequently, a one-to-one comparison between the teacher maps and the intermediate maps simulates transferring knowledge with a fully-connected mapping between teacher and student attention maps.

Two desirable attributes of pointwise convolution take major roles in effective attention knowledge transfer: (1) convolution operation preserves sequential information in each attention map, while (2) allowing fully-connected weighted mapping between the groups.

We can represent the intermediate attention maps, H^I_c, from student attention maps, $H^S_{(m,n)}$ with the learnable parameters of AAM:

$$H^I_c = \sum_{m=1}^{M} \sum_{n=1}^{N} w_{(m,n),c} H^S_{(m,n)} + b_c, (c \in 1, ..., C) \tag{9}$$

Here, $w_{(m,n),c}$ and b_c represent the weight and bias of AAM, respectively. These parameters generate the c-th intermediate attention map ($c \in 1, ..., C$), where C denotes the total number of teacher's attention heads. The variables M and N denote the number of student attention heads within a single layer, and the maximum layer depth of the student respectively.

In summary, AAM is a pointwise convolution layer that generates C intermediate attention maps from a total of $M * N$ student attention maps. The convolution operation of AAM ensures that a one-to-one comparison of intermediate maps to teacher attention maps functions similarly to a fully-connected comparison between student and teacher attention maps. The trained weights of AAM ($w_{(m,n),c}$) after the distillation can be used to analyze the alignment between student and teacher attention heads (discussed further in the Analysis section; see Figure 2).

Finally, the AAM only adds a small number of extra parameters and operations when determining map discrepancy, compared to the student model. These are estimated as $M * N * C$ for parameters and $C * L^2$ for operations. After training, the AAM can be discarded, as it serves no function during inference.

4.2.2. Module Training

AAM generates intermediate attention maps (denoted as $\mathbf{H}_c^I$) that have the same number as the teacher attention maps, referred to as $\mathbf{H}_c^T$. We minimize the KL-Divergence between the attention distributions of the intermediate attention maps and the teacher attention maps as:

$$\mathcal{L}_{att} = \sum_{c=1}^{C} D_{KL}(\mathbf{H}_c^T || \mathbf{H}_c^I). \quad (10)$$

This equation represents the attention transfer loss function, $\mathcal{L}_{\text{att}}$, which quantifies the difference between the teacher's attention maps and the intermediate attention maps. To break down further, $\mathcal{L}_{\text{att}}$ incorporates three different terms according to the types of attention maps.

$$\mathcal{L}_{\text{att}} = \mathcal{L}_{\text{att}}^{\text{enc-self}} + \frac{1}{2}\left(\mathcal{L}_{\text{att}}^{\text{dec-self}} + \mathcal{L}_{\text{att}}^{\text{dec-cross}}\right) \quad (11)$$

$\mathcal{L}_{\text{att}}^{\text{enc-self}}$, $\mathcal{L}_{\text{att}}^{\text{dec-self}}$, and $\mathcal{L}_{\text{att}}^{\text{dec-cross}}$ above correspond to the loss in Equation (10) estimates from three different types of attention maps in Equation (6), (7), (8). To balance the loss scale between the encoder and decoder, we divide the sum of losses for the decoder by 2. To sum up, the attention transfer loss $\mathcal{L}_{\text{att}}$ incorporates three types of attention mechanism in the encoder-decoder model (i.e. self-attention of each encoder and decoder, and cross attention) with balancing for the encoder and decoder-side losses.

Since $\mathcal{L}_{\text{att}}$ is orthogonal to the loss from knowledge distillation by Hinton et al. ($\mathcal{L}_{\text{KD}}$), it can be used jointly to give additional supervision. Throughout the paper, we denote the distillation approach with only $\mathcal{L}_{\text{KD}}$ as vanilla KD. To form a final loss for A2D, we add the cross-entropy loss $\mathcal{L}_{\text{CE}}$ of student model for translation task as follows:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda\mathcal{L}_{\text{att}} + \mu\mathcal{L}_{\text{KD}} \quad (12)$$

The student model and the AAM are collaboratively optimized through an end-to-end approach, as described in Equation (12). Given that H_c^I is differentiable with respect to w_c, the $\mathcal{L}_{\text{att}}$ from Equation (10) actively modifies w_c and the student's parameters to minimize the KL-Divergence between the teacher's attention maps and the intermediate attention maps. At the same time, both $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{KD}}$ indirectly affect the adjustment of w_c, since $\mathbf{H}^S$ is related to the student model's predictions. The hyperparameters λ and μ serve as modulating factors to balance the weights of $\mathcal{L}_{\text{att}}$ and $\mathcal{L}_{\text{KD}}$. In line with Dynamic KD (Li et al., 2021), we modulate the value of λ during training to adjust the supervision derived from $\mathcal{L}_{\text{att}}$ and $\mathcal{L}_{\text{KD}}$.

4.3. Comparison with Previous Methods

Previous works have also utilized attention distributions as knowledge features (Jiao et al., 2020; Wang et al., 2020). However, their approaches treat attention heads within the same layer as a singular unit for distillation. Considering that the attention map from each head captures distinct information across the layer (Voita et al., 2019; Gong et al., 2021), establishing a rigid mapping between student and teacher layers imposes a potential loss of knowledge from the teacher.

In contrast, A2D facilitates a flexible alignment between each individual attention head of the teacher and student models, eliminating the need for pre-defined mapping combinations (Wu et al., 2020) or bucket divisions (Passban et al., 2021). Additionally, A2D is not bound by architectural constraints, such as matching the number of heads or layers, or embedding dimensions, between the teacher and student models.

5. Experiments

5.1. Datasets

We use the public IWSLT and WMT datasets to evaluate our method on translation. The datasets of low-resource scenario include the IWSLT-2014 German $\rightarrow$ English (De $\rightarrow$ En), IWSLT-2017 English $\rightarrow$ Chinese (En $\rightarrow$ Zh), WMT-2022 German $\rightarrow$ Lower Sorbian (De $\rightarrow$ Dsb) translation. Tokenization is done with Subword-NMT (Sennrich et al., 2016) for IWSLT-2014 dataset and Sentencepiece (Kudo and Richardson, 2018) for the others. To prove the effectiveness of our method on high-resource scenarios, we evaluate NMT models on the WMT-2014 English $\rightarrow$ German (En $\rightarrow$ De) datasets. we use newstest2013 datasets as a validation set and newstest2014 as the test set. Data preparation for WMT-2014 En$\rightarrow$De follows (Vaswani et al., 2017) to ensure a fair comparison of baselines in Table 2.

5.2. Distillation Settings

For every experiment, the teacher and student are trained and evaluated with the same datasets.
Low-resource translation Teacher models are 6-layer Transformers (Vaswani et al., 2017) with 4 attention heads, hidden dimensions, and the feed-forward dimension of 512, and 1024 for each. Unless specified otherwise, student models are 3-layer Transformers with the same hyperparameters as the teacher, except for the number of layers. For a fair comparison with CKD (Wu et al., 2020) which reported results with 2-layer student networks, we additionally test our approach for 2-layer Transformer students. To show that A2D does

Models	De → En	De → Dsb	En → Zh
Teacher (6-layer)	36.79 ±0.51	38.68 ±2.02	23.97 ±0.34
Student (3-layer)			
No KD	36.24 ±0.50	35.88 ±1.97	23.20 ±0.31
Vanilla KD (Hinton et al., 2015)	37.00 ±0.50	35.25 ±1.96	24.47 ±0.34
Sequence KD (Kim and Rush, 2016)	36.61 ±0.51	33.41 ±1.76	**24.67** ±0.35
Selective KD (Wang et al., 2021)	37.30 ±0.50	35.94 ±1.95	22.66 ±0.32
TinyBERT (Jiao et al., 2020)	37.24 ±0.52	38.01 ±1.80	24.31 ±0.34
MiniLM (Wang et al., 2020)	36.93 ±0.50	36.43 ±1.97	24.32 ±0.33
ALP-KD (Passban et al., 2021)	37.13 ±0.51	37.07 ±2.05	24.36 ±0.33
A2D (4 heads)	37.17 ±0.49	38.61 ±1.96	24.61 ±0.32
A2D (8 heads)	**37.75** ±0.51	**39.49** ±2.06	24.32 ±0.33
Student (2-layer)			
No KD	35.56 ±0.51	35.89 ±1.88	**23.69** ±0.33
CKD-sc (Wu et al., 2020)	35.73 ±0.49	32.90 ±1.85	22.55 ±0.33
CKD-cc (Wu et al., 2020)	35.46 ±0.48	31.28 ±1.74	22.86 ±0.31
A2D (4 heads)	**36.68** ±0.51	**37.06** ±1.94	23.64 ±0.34

Table 1: BLEU scores of various KD approaches across language pairs. 'No KD' denotes student models trained exclusively with the cross-entropy loss. Each baseline model incorporates 4 attention heads per layer. For a direct comparison with CKD, we train our students with A2D across 2 layers. The highest scores among student models are highlighted in bold, while the second highest are underlined.

Teacher	No KD	PKD	CKD-rc	CKD-oc	**A2D**	A2D (w/o $\mathcal{L}_{\mathrm{KD}}$)
27.70 ±0.65	25.74 ±0.61	23.38	24.14	23.97	**26.37** ±0.64	25.97 ±0.61

Table 2: BLEU scores for WMT-2014 En → De. 'PKD' and 'CKD' refer to Patient KD (Sun et al., 2019) and Combinatorial KD (Wu et al., 2020), respectively. Underlined results are imported from CKD (Wu et al., 2020). Our reproduced teacher and No KD student model, used for A2D training on En → De, yielded slightly better BLEU scores than those reported in the CKD paper, which are 27.03 and 24.31 respectively. Nevertheless, the trend of their model underperforming compared to No KD remains consistent.

not require the same number of attention heads between student and teacher, we also present the results with student models having 8 attention heads. For loss scaling, in Equation (11), λ and μ are initially set to 1, and we set exponential decay on λ at a rate of 0.9 over epochs.

High-resource translation To properly scale the model to the high-resource data, we enlarge the teacher model and student model to 12-layer and 4-layer, respectively. The other hyperparameter settings such as attention heads and hidden dimension are set identically with (Vaswani et al., 2017). To adjust to the increased number of attention maps, We used λ of $\mathcal{L}_{\mathrm{att}}$ as 0.1.

5.3. Baselines

Selective KD (Wang et al., 2021) is a variant of vanilla KD (Hinton et al., 2015) which selectively chooses words to distill based on entropy. For selective KD, we use a "word rate" of 0.5. For Tiny-BERT (Jiao et al., 2020), every other layer of the teacher model is correspondingly mapped to a layer in the student model. MiniLM is originally proposed for knowledge distillation in the pre-training stage,

so we augment the MiniLM objective with $\mathcal{L}_{\mathrm{CE}}$ and $\mathcal{L}_{\mathrm{KD}}$ for a fair comparison. For ALP-KD (Passban et al., 2021), we train the student with an attention mask spanned over all teacher layers. For Combinatorial KD (Wu et al., 2020) (CKD), '-sc', '-cc', '-rc', and '-oc' refer to its different layer mapping configurations.

5.4. Results on Low-resource datasets

Low-resource datasets present a unique challenge for NMT models, emphasizing the importance of effective knowledge distillation. Table 1 summarizes all the results for low-resource NMT. Models are compared with BLEU scores computed using sacreBLEU (Post, 2018), with a confidence interval of 95%, and the number of bootstrap resamples is 1000. Previous KD methods, such as TinyBERT and MiniLM, impose a constraint that the number of heads in the student must match that of the teacher. In contrast, A2D does not have this limitation regarding the number of attention heads, allowing us to train students with 8 heads using teachers with only 4 heads. We also present the results of students with 8 heads in section 6. Doubling the

number of attention heads from 4 to 8 does not alter the overall parameter count in the 'A2D (8heads)' model. This is achieved by proportionally reducing the dimension per head ($d_{head} = d_{model}/n_{head}$) while keeping the feature dimension (d_{model}) constant, thus ensuring a fair comparison.

In the De → En and De → Dsb language pairs, students trained with A2D yield higher BLEU scores than those trained with other KD methods. For the En → Zh language pair, A2D yields superior results compared to other feature-based KD baselines and is on par with Sequence-level KD. In all language pair settings, students trained with A2D surpass the performance of their teachers despite having only half the number of transformer layers. Remarkably, the effectiveness of our method was most evident in the De → Dsb dataset, which had the least amount of training data at 39K samples. These results indicate that training with A2D allows students to achieve better generalization, particularly with low-resource training data.

To compare with CKD, we trained 2-layer students using A2D. While the score of CKD varies on its mapping option, our model consistently outperforms CKD regardless of the dataset or their mapping option. Moreover, a 2-layer student in De → En achieves results comparable to its 6-layer teacher. The results not only validate the robustness of A2D but also show that our learned alignment of attention heads is more effective for knowledge transfer than the heuristic mapping of CKD.

In low-resource settings, a student model could outperform the teacher under the guidance of the teacher model. This is due to the enhanced generalization provided by the regularization effect of distillation, as discussed in (Mobahi et al., 2020; Yuan et al., 2020). The teacher model, often being larger and more complex, captures rich feature representations (such as attention maps) of the data. Through distillation, the student model learns these representations, which might be more generalizable than those learned from the relatively small amount of raw data alone.

5.5. Results on High-resource dataset

To assess the versatility of our method, we extended our experiments to the high-resource dataset. Table 2 describes our results with different baselines. From our observations, traditional KD techniques like Patient KD and Combinatorial KD did not enhance the performance of the student models for high-resource data. Surprisingly, they even underperform compared to a student model trained without any KD techniques.

However, a noticeable distinction arises when training the student model with the A2D approach, demonstrating a performance level similar to that

n_{head}	A2D	No KD	ΔBLEU	$\mathcal{L}_{\text{att}}$
2	36.33	35.79	0.54	0.036
4	37.17	36.24	0.93	0.024
8	**37.75**	**36.25**	1.5	0.016
16	37.19	35.78	1.41	0.013

Table 3: BLEU score and our attention distillation loss ($\mathcal{L}_{\text{att}}$) at convergence over a different number of heads (n_{head}) in A2D and No KD. ΔBLEU indicates BLEU score difference between A2D and No KD.
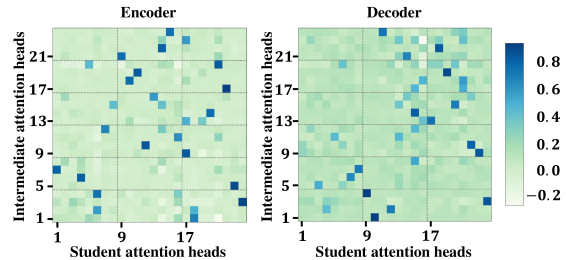

Figure 2: Attention head connection weights in the trained AAM. Axes indicate attention head numbers in the student (3 layers of 8 heads) and teacher (6 layers of 4 heads) models. The dashed grid shows layer boundaries. Darker colors signify stronger connections. Best viewed in color.

of its teacher, even when modeled with only one-third number of the transformer layers compared to its teacher. This performance gain remains even without integrating our method with vanilla KD, underscoring its effectiveness.

6. Analysis and Discussion

6.1. Effect of Fine-grained Alignment

We focus on two attributes of A2D that enable fine-grained alignment: (1) dense, head-wise distillation (2) the use of an attention map as a feature. Our hypothesis is that the detailed head-wise attention alignment is what gives A2D-trained students an edge in performance. In Table 3, we present how our attention transfer loss and performance gap between students trained with A2D and students trained without KD vary with different numbers of attention heads at the point of convergence. Note that varying the number of heads in the student model does not change the total number of parameters. Our observation suggests that as we reduce the $\mathcal{L}_{\text{att}}$, ΔBLEU becomes more pronounced, which indicates the effectiveness of our method. This inverse correlation between $\mathcal{L}_{\text{att}}$ and ΔBLEU as the number of heads increases could be attributed to our attention alignment module (AAM), which generates intermediate attention maps from original

Datasets	A2D (All)	A2D (Enc-Self)	A2D (Dec-All)	A2D (Dec-Self)	A2D (Dec-Cross)
De → En	**37.75** ±0.51	37.39 ±0.50	37.21 ±0.51	37.16 ±0.49	37.25 ±0.51
De → Dsb	**39.49** ±2.06	36.80 ±2.01	37.81 ±1.95	36.81 ±2.13	37.66 ±1.97
En → Zh	24.32 ±0.33	**24.77** ±0.32	24.75 ±0.35	24.67 ±0.33	24.64 ±0.35

Table 4: Ablation studies by applying A2D on different parts of Transformer. Dec-All setting indicates that both self-attention and cross-attention maps are used for A2D. For Dec-Self and Dec-Cross settings, we increased the weight of $L_{\mathrm{att}}^{\mathrm{dec-self}}$ and $L_{\mathrm{att}}^{\mathrm{dec-cross}}$ from $1/2$ to 1 respectively to match the loss scale on the Dec-All setting.

Model	#Params	CoLA (Mcc)	MNLI-(m/mm) (Acc)	SST-2 (Acc)	QNLI (Acc)	MRPC (F1)	QQP (Acc)	RTE (Acc)	STS-B (Spear)	Avg
BERT$_{base}$	110M	58.7	84.5/84.5	91.7	91.3	89.0	91.1	67.9	89.5	82.9
BERT$_6$	66M	51.2	81.7/82.6	91.0	89.3	89.2	90.4	66.1	88.3	80.9
PD	66M	-	82.5/83.4	91.1	89.4	**89.4**	90.7	66.7	-	-
PKD	66M	45.5	81.3/-	91.3	88.4	85.7	88.4	66.5	86.2	79.2
TinyBERT	66M	53.8	83.1/83.4	**92.3**	89.9	88.8	90.5	66.9	88.3	81.7
A2D	66M	**58.8**	**83.2/83.5**	91.7	**90.3**	89.2	**90.9**	**67.5**	**88.7**	**82.5**

Table 5: Evaluation results on the dev set of GLUE Benchmark. We use BERT$_{base}$ and BERT$_6$ as teacher and student model, respectively. Both BERT$_6$, a 6-layer smaller variant of BERT, and 'PD', a distilled model, are released by Turc et al.. The results of baselines are imported from Park et al..

Datasets	Layer-wise A2D	Head-wise A2D (Default)
De → En	37.01 ±0.49	**37.75** ±0.49
De → Dsb	36.70 ±1.96	**39.49** ±2.06
En → Zh	24.07 ±0.34	**24.32** ±0.33

Table 6: Comparison of our original A2D (head-wise distillation) with its layer-wise counterpart.

student maps. AAM draws *all* the student attention maps ($\mathbf{H}^S$) to mimic *each* teacher map ($\mathbf{H}^T_c$) via pointwise convolution operation; approximating $\mathbf{H}^T$ from an increased number of $\mathbf{H}^S$ is more feasible. However, employing an excessive number of heads might degrade the performance by reducing the expressiveness of each student's attention head. We hypothesize that the choice of $n_{head} = 16$ was overly complex for a model with specifications $d_{model} = 512$ and reduced the capacity of each head. This led to the model's optimal performance in BLEU with $n_{head} = 8$, but recorded a dip in performance when escalated to $n_{head} = 16$.

In Figure 2, we present the connectivity between attention maps learned by AAM (pointwise convolution layer) with heatmap. It shows that heads from students associate not only within but across the layers to form intermediate maps ($\mathbf{H}^I$) that are purposed to emulate the teacher maps ($\mathbf{H}^T$). This suggests that transferring knowledge using entire layers as units may not be the most effective approach for knowledge transfer. To justify the claim, we also carried out evaluations on the layer-wise variant of A2D, as demonstrated in Table 6. Layer-wise A2D, which uses per-layer averaged maps as

a knowledge feature, underperforms the original head-wise A2D by a significant margin on every language pair. This observation reinforces our hypothesis emphasizing the impact of our head-wise comparison approach in distillation.

6.2. Decoder Distillation

While previous feature-based KD methods focused on distilling encoder-only models (Jiao et al., 2020; Wang et al., 2020; Passban et al., 2021) and did not discover effective KD settings for the decoder in NMT tasks (Wu et al., 2020), A2D distills both the encoder and decoder.

Our claim regarding the effectiveness of our method on the decoder is supported by the ablation studies presented in Table 4. Generally, applying A2D to both the encoder and decoder together yielded the best results. In the En → Zh direction, there is a slight performance degradation when using integrated encoder and decoder distillation compared to using encoder-only or decoder-only distillation. Nevertheless, our model trained with decoder-only distillation outperforms the one with distillation on both the encoder and decoder, demonstrating its effectiveness on the decoder.

To investigate why A2D is effective on the decoder, we examine the AAM of the decoder, as described in Figure 2. From the heatmap, we observe that the connection between heads tends to be sparse at the encoder level, with most values near 0. Conversely, in the decoder, the connection between heads is more evenly distributed, with values shifting away from 0 and closer to 0.2. Based on this observation, we believe that our soft,

fine-grained connections between teacher-student features led to more successful KD for decoder tasks, as compared to the strict on/off connections proposed in earlier KD studies.

6.3. Effectiveness on Different Tasks

In this study, we assessed our method using NMT to demonstrate its potential beyond tasks solely associated with encoders. Additionally, we present performance metrics for the A2D method when applied to encoder-only models on natural language understanding benchmarks given that most of the KD in natural language processing research has focused on the encoder-only models and tasks. Specifically, we applied our method to BERT (Devlin et al., 2019) distillation, benchmarked on the GLUE (Wang et al., 2018) dataset. The comparative results with baselines are presented in Table 5. For training, we used fine-tuned teacher models for each GLUE task and then applied A2D to the corresponding student model. We experimented with the hyperparameter λ of Equation (12), selecting from the values {0.01, 0.02, 0.05, 0.1}. For all other configurations, we followed BERT's settings. Notably, our method demonstrated superior performance over encoder-oriented baselines, even without dedicated task-specific hyperparameter tuning.

7. Conclusion

In this paper, we introduce Align-to-Distill, a novel approach to knowledge distillation that enables a detailed alignment of attention heads between teacher and student models. We propose a strategy to overcome the need for heuristic feature mapping in a learnable manner. Our approach shows promising results in decoder distillation, effectively compressing models while preserving translation quality.

8. Limitations and Future work

Although A2D is architecturally flexible without constraints on hidden size or number of attention heads, the teacher and student models must share the same vocabulary. This requirement potentially restricts its broader applicability. Also, while our work demonstrates A2D's effectiveness on the decoder module in Table 4, we have not yet tested A2D on decoder-only models. The scope of this paper primarily focuses on encoder-decoder-based translation models. In future work, we plan to extend A2D's application to decoder-only models. Moreover, the concept of A2D may encompass a broader range of architectures in future work, as the idea of adaptively aligning features is not restricted to using attention as a feature.

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, (NIPS)*, pages 2654–2662.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations, (ICLR) Conference Track Proceedings*.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of International Conference on Knowledge Discovery and Data Mining, (SIGKDD)*, pages 535–541.

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL-HLT)*, pages 4171–4186.

Hongyu Gong, Yun Tang, Juan Pino, and Xian Li. 2021. Pay better attention to attention: Head selection in multilingual and multi-domain sequence modeling. *Advances in Neural Information Processing Systems*, 34:2668–2681.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *Int. J. Comput. Vis.*, 129(6):1789–1819.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1053–1062. Association for Computational Linguistics.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: (EMNLP)*, pages 4163–4174.

James M Joyce. 2011. Kullback-leibler divergence. In *International encyclopedia of statistical science*, pages 720–722. Springer.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. 2020. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *International Conference on Learning Representations*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, (EMNLP)*, pages 66–71.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, (ACL)*, pages 7871–7880.

Lei Li, Yankai Lin, Shuhuai Ren, Peng Li, Jie Zhou, and Xu Sun. 2021. Dynamic knowledge distillation for pre-trained language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 379–389.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pre-training approach. *CoRR*, abs/1907.11692.

Hossein Mobahi, Mehrdad Farajtabar, and Peter L. Bartlett. 2020. Self-distillation amplifies regularization in hilbert space. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Geondo Park, Gyeongman Kim, and Eunho Yang. 2021. Distilling linguistic context for language model compression. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 364–378. Association for Computational Linguistics.

Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. 2021. ALP-KD: attention-based layer projection for knowledge distillation. In *Conference on Innovative Applications of Artificial Intelligence, (IAAI), The Symposium on Educational Advances in Artificial Intelligence, (EAAI)*, pages 13657–13665.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191. Association for Computational Linguistics.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations, (ICLR)*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, (ACL)*.

Sam Shleifer and Alexander M. Rush. 2020. Pre-trained summarization distillation. *CoRR*, abs/2010.13002.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing, (EMNLP-IJCNLP)*, pages 4322–4331.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, (ACL)*, pages 2158–2170.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, (NIPS)*, pages 3104–3112.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv: Computation and Language*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez,

Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, (NIPS)*, pages 5998–6008.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multihead self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355. Association for Computational Linguistics.

Fusheng Wang, Jianhao Yan, Fandong Meng, and Jie Zhou. 2021. Selective knowledge distillation for neural machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing, (ACL/IJCNLP)*, pages 6456–6466.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, (NeurIPS)*.

Yimeng Wu, Peyman Passban, Mehdi Rezagholizadeh, and Qun Liu. 2020. Why skip if you can combine: A simple knowledge distillation technique for intermediate layers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, (EMNLP)*, pages 1016–1021.

Li Yuan, Francis E. H. Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2020. Revisiting knowledge distillation via label smoothing regularization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 3902–3910. Computer Vision Foundation / IEEE.

Sergey Zagoruyko and Nikos Komodakis. 2017. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations, (ICLR)*.

Jiawei Zhou, Jason Eisner, Michael Newman, Emmanouil Antonios Platanios, and Sam Thomson. 2022. Online semantic parsing for latency reduction in task-oriented dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1576. Association for Computational Linguistics.