

# Efficient Speech Translation through Model Compression and Knowledge Distillation

Yasmin Moslem

ADAPT Centre

School of Computer Science and Statistics

Trinity College Dublin

Dublin, Ireland

## Abstract

Efficient deployment of large audio-language models for speech translation remains challenging due to their significant computational requirements. In this paper, we address this challenge through our system submissions to the “Model Compression” track at the International Conference on Spoken Language Translation (IWSLT 2025). We experiment with a combination of approaches including iterative layer pruning based on layer importance evaluation, low-rank adaptation with 4-bit quantization (QLoRA), and knowledge distillation. In our experiments, we use Qwen2-Audio-7B-Instruct for speech translation into German and Chinese. Our pruned (student) models achieve up to a 50% reduction in both model parameters and storage footprint, while retaining 97-100% of the translation quality of the in-domain (teacher) models.

## 1 Introduction

Multimodal foundation models have shown powerful capabilities in different tasks, including speech translation. However, these models are often large and computationally intensive, making them impractical to use in real-world settings with limited resources. To enhance the efficiency of these models, researchers have been investigating diverse approaches to model compression that aim to reduce the computational requirements while retaining performance (Gandhi et al., 2023; Peng et al., 2023b,a; Treviso et al., 2023; Wang et al., 2023).

Qwen2-Audio (Chu et al., 2023, 2024) is a state-of-the-art foundation model that accepts various audio signal inputs and performs audio analysis or direct textual responses to speech instructions. In the IWSLT 2025’s Model Compression track (Abdulmumin et al., 2025), the organizers required that all submissions must be derived from the Qwen2-Audio model. The official languages of the task are English-to-German (EN-DE) and English-to-Chinese (EN-ZH).

We experimented with various approaches including efficient fine-tuning using quantized low-rank adapters with 4-bit quantization (QLoRA) (Hu et al., 2021; Dettmers et al., 2023), iterative layer pruning based on layer importance evaluation (Peer et al., 2022; Gandhi et al., 2023; Sajjad et al., 2023), and sequence-level knowledge distillation (Kim and Rush, 2016; Crego and Senellart, 2016; Jooste et al., 2022; Gandhi et al., 2023). Our experiments are mainly based on *Qwen2-Audio-7B-Instruct*.<sup>1</sup> While Section 3 elaborates on our experiments, we can summarize our two submissions for German and Chinese as follows:

- **Setup 1:** This model is the outcome of fine-tuning *Qwen2-Audio-7B-Instruct* in two stages: (a) full fine-tuning with the ACL 60/60 dataset, and (b) QLoRA fine-tuning with 4-bit quantization using the ACL 60/60 dataset augmented with data knowledge distillation from the fully fine-tuned model. This process has achieved 40% compression in terms of both model parameters and storage size. We discuss the details of this model in Section 3.2.
- **Setup 2:** This model is a pruned version of *Qwen2-Audio-7B-Instruct*, and was created in multiple stages: (a) full fine-tuning of the baseline model with the ACL 60/60 dataset, (b) layer pruning of the decoder into 24 layers, while all 32 encoder layers were kept intact, (c) full fine-tuning of the pruned model, and (d) QLoRA fine-tuning with the ACL 60/60 dataset augmented with data knowledge distillation from the fully fine-tuned model and a portion of the CoVoST2 dataset to restore the quality of the teacher model. This process has achieved 50% compression in terms of both model parameters and storage size. We discuss the details of this model in Section 3.3.

<sup>1</sup><https://hf.co/Qwen/Qwen2-Audio-7B-Instruct>

Model	EN-DE			EN-ZH			Params	Storage
	BLEU	chrF++	COMET	BLEU	chrF	COMET	(B)	(GB)
Baseline	22.96	49.88	8.61	38.73	32.53	17.38	8.40	16.79
+ Full Fine-tuning	39.28	65.27	56.32	58.54	52.54	65.97		
+ QLoRA Fine-tuning	41.52	66.25	53.47	57.65	51.08	65.77	<b>4.95</b>	<b>9.64</b>
+ Knowledge Distillation	<b>43.25</b>	<b>68.02</b>	<b>59.36</b>	<b>59.60</b>	<b>53.94</b>	<b>68.42</b>		

Table 1: Evaluation of the experiment that employs QLoRA with 4-bit quantization and augments the authentic data with knowledge distillation. This approach reduces the model size by more than 40% in terms of both the number of parameters (params) and storage footprint, while achieving the best translation performance for both English-to-German (EN-DE) and English-to-Chinese (EN-ZH) language pairs.

## 2 Data

### 2.1 In-Domain Data

The ACL 60/60 dataset<sup>2</sup> (Salesky et al., 2023) is used in all of our experiments as the in-domain data. ACL 60/60 contains multilingual translation of ACL 2022 technical presentations into 10 target languages. The dataset consists of two splits, “dev” and “eval”, which together comprise 884 utterances. We merged the two splits, and randomly sampled 100 utterances for testing, which left us with 784 utterances for training. For test data sampling, we used the *train\_test\_split* method from the *datasets*<sup>3</sup> library (Lhoest et al., 2021), setting the random seed option to zero. The ACL 60/60 dataset was required for “constrained” submissions to the IWSLT’s “Model Compression” track.

### 2.2 Out-of-Domain Data

After layer pruning of the *Qwen2-Audio-7B-Instruct* model (cf. Section 3), we needed to use more training data to restore the translation quality of the unpruned model. In addition to knowledge distillation data from the teacher model, we used a portion of the CoVoST2 dataset (Wang et al., 2021) (cf. Section 3.3). CoVoST2 is a large-scale multilingual speech-to-text translation corpus, covering translations from English into 15 languages, including German and Chinese.

## 3 Methodologies

We experiment with diverse methods to compress *Qwen2-Audio-7B-Instruct* while maintaining the translation quality. This section covers our two main experimental setups, while Section 4 dis-

cusses several ablation studies and elaborates on intermediate experiments.

Our first experimental setup (cf. Section 3.2) employs QLoRA fine-tuning with 4-bit quantization, while the other experimental setup (cf. Section 3.3) conducts layer importance evaluation and applies iterative layer pruning of the model, followed by QLoRA fine-tuning. Both setups use knowledge distillation to recover the translation quality of the in-domain teacher model.

### 3.1 Full-Parameter Fine-tuning

In all of our experiments, we start by full-parameter fine-tuning of the *Qwen2-Audio-7B-Instruct* model on the ACL 60/60 dataset. This step is essential to ensure the foundation model is familiar with the downstream task and in-domain data. In particular, we train the baseline model on the in-domain data for 3 epochs, using a batch size of 4, learning rate of 1e-5, weight decay of 0.001, and no warm-up steps. The model is initially loaded in *bfloat16* data type. As shown in Table 1 and Table 2, the fully fine-tuned model clearly outperforms the baseline by an average of 48 COMET points for translation into German and Chinese. Hence, this fully fine-tuned model is used as a foundation for both our experiments in Section 3.2 and Section 3.3. It is also used as a “teacher” for knowledge distillation into the compressed “student” models. We conduct the training on one H200 SXM GPU, using the Transformers framework<sup>4</sup> (Wolf et al., 2020).

### 3.2 QLoRA with 4-bit Quantization

This experimental setup employs quantization, accompanied by efficient fine-tuning, and knowledge distillation techniques. We start with full-parameter

<sup>2</sup><https://hf.co/datasets/yoslem/acl-6060>

<sup>3</sup><https://github.com/huggingface/datasets>

<sup>4</sup><https://github.com/huggingface/transformers>

Model	Data	EN-DE			EN-ZH			Params	Storage
		BLEU	chrF++	COMET	BLEU	chrF	COMET	(B)	(GB)
Baseline	-	22.96	49.88	8.61	38.73	32.53	17.38	8.40	16.79
+ Full FT	ACL	39.28	65.27	56.32	58.54	52.54	65.97		
+ Pruning	ACL	10.78	39.58	-44.4	42.52	36.92	39.42	6.78	13.55
+ FT	ACL	32.16	60.39	39.08	53.05	47.23	56.72		
	+ KD	33.44	60.91	39.23	53.41	48.20	54.94		
+ QLoRA	+ CV	39.59	65.14	59.21	56.52	50.74	64.34	<b>4.12</b>	<b>8.65</b>

Table 2: Evaluation of the iterative layer pruning experiment. We started by full-parameter fine-tuning (Full FT) of the baseline model *Qwen2-Audio-7B-Instruct* on the in-domain dataset ACL 60/60. Pruning 8 layers of the decoder of the model achieved approx. 20% reduction in the model size; however, it affected the quality of the model. Hence, we fine-tuned the pruned model again on the in-domain dataset to restore as much as possible of the quality of the fully fine-tuned model. Finally, we fine-tuned the resulting model with low-rank adaptation after quantizing it into the 4-bit precision (QLoRA) on a mix of the in-domain data, knowledge distillation data (KD) and out-of-domain data, namely the CoVoST2 (CV) dataset. The whole process of pruning followed by QLoRA fine-tuning with 4-bit quantization has resulted in approx. 50% reduction in the model size, while retaining 97% and 100% of the translation quality for Chinese and German, respectively, compared to the teacher model.

fine-tuning of *Qwen2-Audio-7B-Instruct* (cf. Section 3.1). Afterwards, the fine-tuned model is quantized into the 4-bit precision and then fine-tuned with low-rank adapters (QLoRA) (Hu et al., 2021; Dettmers et al., 2023). Moreover, we use the fully fine-tuned model as a “teacher” in the knowledge distillation process.

**Knowledge distillation:** To restore the quality of the in-domain fully fine-tuned teacher models (cf. Section 3.1), sequence-level knowledge distillation is applied (Kim and Rush, 2016; Gandhi et al., 2023). In other words, we translate the ACL 60/60 training data with the “teacher” model. The training data is then augmented with the knowledge distillation data, and duplicates are filtered out. As a result, the augmented data after knowledge distillation comprises 1,568 segments for German and 1,069 segments for Chinese. Finally, “student” models are fine-tuned with QLoRA on the augmented data.

**QLoRA fine-tuning:** First, we enable 4-bit quantization through *BitsAndBytes* (Dettmers et al., 2023) while loading the in-domain fully fine-tuned model. In the configuration of *BitsAndBytes*, we set the quantization type to “nf4”, and use the “double\_quant” option, where the quantization constants from the first quantization are quantized again. For LoRA configuration, we set the rank to 64, alpha to 128, and dropout to 0. We target all linear modules. Overall, this configuration results

in 2.41% trainable parameters of the model. Moreover, we enable Rank-Stabilized LoRA (rsLoRA) (Kalajdzievski, 2023). We train the model for 4 epochs, using a batch size of 4, and a learning rate of 1e-5.

As Table 1 illustrates, the combination of QLoRA with 4-bit quantization and knowledge distillation has achieved the highest translation performance into both German and Chinese across all evaluation metrics while reducing the model size in terms of both the number of parameters and storage requirements by more than 40% compared to the baseline model.<sup>5</sup>

### 3.3 Iterative Layer Pruning

In this experimental setup, we apply iterative layer pruning to the fully fine-tuned “teacher” model (cf. Section 3.1). This approach incrementally identifies and removes layers with minimal contribution to translation quality, one layer at a time. The pruned model resulting from this process is then fine-tuned on both the ACL 60/60 training dataset and knowledge distillation data from the teacher model. Finally, the model is further fine-tuned with QLoRA, leveraging 4-bit quantization for efficient

<sup>5</sup>For consistency, our storage footprint calculations are based only on the size of model files (\*.safetensors), including the adapter of QLoRA models. We exclude the tokenizer and configuration files, as their contribution to the overall size is relatively small (~ 13 MB) and they are unaffected by our optimization methods. All sizes are computed using the decimal definition of gigabytes, where 1 GB = 1000<sup>3</sup> bytes.

low-rank adaptation. Pruning 8 layers achieves a 20% reduction in model size, which increases to 50% when combined with 4-bit quantization. Fine-tuning the pruned model restores between 97% and 100% of the teacher model’s translation quality for Chinese and German, respectively. The following points elaborate on the process.

**Layer importance evaluation:** We conduct layer importance evaluation by measuring translation performance without each layer. In this greedy layer pruning approach (Peer et al., 2022; Rostami and Dousti, 2024), to prune  $n + 1$  layers, only a single optimal layer to prune must be added to the already known solution for pruning  $n$  layers. After identifying and removing the least critical layer, we repeat the layer importance evaluation on the remaining layers until reaching our  $n$  pruning target. We observe that while removing certain layers of the model (e.g. the first or last layer) substantially degrades translation performance, others result in minimal performance drops. When experimenting with using either the chrF/chrF++ or COMET metric for layer importance evaluation, the models pruned based on chrF/chrF++ outperform those pruned based on COMET.

**Layer pruning:** We iteratively prune one decoder layer at a time, selecting the layer whose removal has the least negative impact on translation quality, measured by chrF/chrF++ scores. At each iteration, we evaluate the translation performance of the pruned model on the test split of the ACL 60/60 dataset, after removing each candidate layer. The layer whose removal yields the best performance is eventually pruned. This process continues until a predefined number of layers (8 in the main experiments) have been removed. By iteratively removing the least important layers, this performance-guided method produces a more compact model that can be fine-tuned further to recover the translation quality of the teacher model. We observe that the performance of the English-to-German model is more impacted by pruning than the English-to-Chinese model, which might be attributed to the pre-training process (cf. Table 2).

**Knowledge distillation:** Knowledge distillation is the process of transferring knowledge from a large model (teacher) to a smaller one (student). In our case, the teacher is the fully fine-tuned model, and the student is the model resulting from iterative pruning. We translate the in-domain data with

the teacher model to augment the authentic data. As the process can result in duplicate translations, we remove duplicate segments from the training data. This step is similar to what we did in the first experimental setup (cf. Section 3.2).

**Fine-tuning:** The pruning step is followed by fine-tuning the pruned model for 4 epochs using the in-domain ACL 60/60 dataset augmented with the knowledge distillation data (Kim et al., 2023; Gandhi et al., 2023). This step recovers most of the translation quality of the teacher model.

**QLoRA fine-tuning:** This step serves two purposes, improving both the compression level and translation performance of the pruned model. After quantizing the model resulting from the previous step, low rank adaptation is used for fine-tuning it further. The training data consists of the ACL 60/60 dataset augmented with the knowledge distillation data from the fully fine-tuned “teacher” model (oversampled by a factor of 10), as well as a portion of the CoVoST2 dataset (100k utterances). We fine-tune the model for 1 epoch, using a batch size of 8 (to make use of the computing resources, since the model is much smaller now), and a learning rate of  $1e-5$ .

This whole process of iterative layer pruning followed by fine-tuning has achieved up to 50% compression in terms of both model parameters and storage size.<sup>6</sup> Moreover, the quality degradation caused by pruning has been mitigated through multi-stage fine-tuning on diverse data. As demonstrated by Table 2, by the end of the process, the pruned model could recover most of the translation quality of the fully fine-tuned teacher model (97% for Chinese and 100% for German).

Our ablation study (cf. Section 4) demonstrates that iterative layer pruning considerably outperforms fixed middle-layer pruning (cf. Section 4.3). Moreover, it clarifies that pruning exclusively decoder layers yields better performance than pruning both encoder and decoder layers (cf. Section 4.2). While our main experiments in this section prune only 8 layers, resulting in a model with 24 decoder layers and 32 encoder layers, the ablation study investigates pruning up to 16 layers (cf. Section 4.5).

<sup>6</sup>To achieve these compression gains from layer pruning, the Qwen2-Audio model must initially be loaded in *bfloat16* data type. For other tasks like fine-tuning and inference, *bfloat16* precision is not required, although it may be necessary when computing resources are limited, potentially at the cost of reduced quality.



Encoder	Decoder	BLEU	chrF++	COMET	Params	Storage
24 ↓	24 ↓	26.44	54.67	13.90	6.62 B	13.24 GB
32 =	24 ↓	<b>30.81</b>	<b>58.45</b>	<b>31.95</b>	6.78 B	13.55 GB

Table 3: Comparison of layer pruning of both the encoder and decoder with layer pruning of the decoder only. Both models are fine-tuned before and after layer pruning on the EN-DE ACL 60/60 dataset. This experiment uses middle layer pruning. The model that prunes the layers of only the decoder outperforms the model that prunes both the encoder and decoder, although the former has a slightly higher number of parameters and storage size.

## 4 Ablation Study

This section elaborates on some intermediate experiments that led us to the final models in Section 3.2 and Section 3.3. These experiments include comparing the performance of the Qwen2-Audio base model with Qwen2-Audio-Instruct (cf. Section 4.1), comparing encoder-decoder pruning with decoder-only pruning (cf. Section 4.2), comparing “iterative” layer pruning with fixed middle-layer pruning (cf. Section 4.3), iterative pruning of up to 16 layers (cf. Section 4.5), fine-tuning before and after pruning (cf. Section 4.6), and using different sizes of out-of-domain data to fine-tune the pruned models (cf. Section 4.7).

### 4.1 Qwen2-Audio Base vs Instruct

We experimented with both *Qwen2-Audio-7B* and *Qwen2-Audio-7B-Instruct* for English-to-German speech translation. As Table 4 the “Instruct” model outperforms its base version. Hence, we use *Qwen2-Audio-7B-Instruct* in all of our experiments throughout the paper.

We follow the prompt requirements of the *Qwen2-Audio* models. For the base model, Qwen2-Audio-7B, the prompt is as follows:

```
"<audio_bos><|AUDIO|><audio_eos>Translate the
English speech into {language}:"
```

For the instruction-following model, *Qwen2-Audio-7B-Instruct*, the prompt is as follows:

```
[
  {"role": "system", "content": "You are a professional
translator."},
  {"role": "user", "content": [
    {"type": "audio", "audio_url": audio_path},
    {"type": "text", "text": "Translate the English
speech into {language}:"}],
  ]
```

Language	Model	BLEU	chrF/++	COMET
EN-DE	base	6.15	32.10	-58.34
	instruct	<b>22.96</b>	<b>49.88</b>	<b>8.61</b>
EN-ZH	base	7.23	10.62	-52.06
	instruct	<b>38.73</b>	<b>32.53</b>	<b>17.38</b>

Table 4: Evaluation of the Qwen2-Audio-7B (base) and Qwen2-Audio-7B-Instruct (instruct) models before fine-tuning. The “instruct” model outperforms the “base” model for both English-to-German (EN-DE) and English-to-Chinese (EN-ZH) speech translation.

### 4.2 Encoder-Decoder Layer Pruning

The Qwen2-Audio is based on the encoder-decoder Transformer architecture (Vaswani et al., 2017). It consists of an encoder for audio (*audio\_tower*) and a decoder for text generation (*language\_model*), each of which comprises 32 layers. We first experimented with layer pruning of both the encoder and decoder. However, inspired by the Distil-Whisper work (Gandhi et al., 2023), we experimented with pruning decoder layers only, which achieved better results. In other words, pruning only the decoder from 32 layers to 24 layers outperformed pruning both the encoder and decoder into 24 layers.

In this experiment, we pruned 8 fixed middle layers, from the 12th to the 19th layer, inclusively. After fine-tuning both models with the English-to-German ACL 60/60 dataset, the model where only the decoder layers were pruned achieved 4+ additional points in terms of BLEU and chrF++ and 18+ points of COMET. It is worth noting that both the number of parameters and the storage footprint of this model is only 1% larger than the model with both the encoder and decoder were pruned. Table 3 shows the performance evaluation results of fine-tuning the English-to-German model after middle-layer pruning.

Language	Pruning	Metric	Pruned Layers	Model	BLEU	chrF/++	COMET
EN-DE	Middle	n/a	[12, 13, 14, 15, 16, 17, 18, 19]	Pruned	0.13	6.13	-162.44
				+ FT	30.81	58.45	31.95
	Iterative	COMET	[12, 1, 9, 15, 20, 27, 29, 5]	Pruned	6.53	31.15	-41.96
				+ FT	30.97	59.67	34.39
		chrF++	[13, 3, 20, 9, 29, 1, 19, 27]	Pruned	10.78	39.58	-44.40
				+ FT	<b>32.16</b>	<b>60.39</b>	<b>39.08</b>
EN-ZH	Middle	n/a	[12, 13, 14, 15, 16, 17, 18, 19]	Pruned	1.3	3.8	-94.05
				+ FT	45.84	40.48	40.37
	Iterative	COMET	[7, 9, 24, 3, 28, 6, 15, 18]	Pruned	21.1	27.25	28.26
				+ FT	51.85	46.36	<b>58.64</b>
		chrF	[7, 25, 3, 4, 29, 15, 26, 20]	Pruned	42.52	36.92	39.42
				+ FT	<b>53.05</b>	<b>47.23</b>	<u>56.72</u>

Table 5: Comparison between **middle-layer pruning** and **iterative layer pruning**, with either COMET or chrF/chrF++ as the metric for measuring layer importance. Iterative layer pruning, i.e. removing layers one by one, and then evaluating the resulting model, outperforms middle-layer pruning. In particular, when chrF/chrF++ is used for layer importance evaluation, the resulting pruned model achieves better speech translation quality after fine-tuning on the ACL 60/60 dataset.

### 4.3 Iterative vs Middle Layer Pruning

In this section, we compare two common approaches to layer-wise pruning, namely iterative layer pruning and middle-layer pruning. Moreover, we compare using chrF/chrF++ and COMET for layer importance evaluation during iterative layer pruning. In all cases, we only work on decoder layers (cf. Section 4.2).

As discussed in Section 3.3, we experimented with iterative pruning based on layer importance evaluation to identify and remove the layers that contribute least to translation quality. In contrast, in middle-layer pruning, we simply remove the 8 middle layers of the model, namely layers 12 through 19 out of the 32 decoder layers of the model.

Since the bottom layers of a model are closer to the input and the top layers are closer to the output, it is possible that both the top layers and the bottom layers are more important than the middle layers. In practice, the impact on model performance after pruning the middle layers varies across different models (Sajjad et al., 2023).

In iterative layer pruning based on performance evaluation (using chrF++ for German and chrF for Chinese), the pruned layers are [1, 3, 9, 13, 19, 20, 27, 29] for German, and [3, 4, 7, 15, 20, 25, 26, 29] for Chinese. This shows a diverse layer selection that is not concentrated solely in the mid-

dle. As Table 5 illustrates, iterative layer pruning yields much better results than middle-layer pruning. For example, when pruning 8 middle layers of the English-to-Chinese model, the final evaluation scores of the resulting model are so low across all metrics (BLEU: 1.3, chrF: 3.8, COMET -94.05), compared to the scores achieved when the same number of layers is iteratively removed based on layer importance (BLEU: 42.52, chrF: 36.92, COMET 39.42).

Moreover, we experimented with both using chrF/chrF++ and COMET for evaluating pruned models during the iterative process. Interestingly, the final model obtained by using chrF/chrF++ for layer importance evaluation achieves better results (cf. Table 5). This might be due to the scientific nature of the ACL 60/60 dataset, and it requires future exploration with other datasets.

### 4.4 Immediate Recovery

In our main experiments, we fine-tuned the pruned models only after completing the entire pruning process (cf. Section 3.3). In order to understand the effect of accompanying iterative pruning with iterative recovery (Wibowo et al., 2025), we conducted an extra experiment where we immediately fine-tuned the model after each layer pruning iteration. In other words, when pruning 8 layers, the

Model	Layers	EN-DE			EN-ZH			Params	Storage
		BLEU	chrF++	COMET	BLEU	chrF	COMET	(B)	(GB)
Pruned [8] + Fine-tuned	24/32	10.78 32.16	39.58 60.39	-44.4 39.08	42.52 53.05	36.92 47.23	39.42 56.72	6.78	13.55
Pruned [10] + Fine-tuned	22/32	4.54 30.90	26.33 59.80	-114.73 33.74	16.51 51.97	23.02 47.19	5.89 59.57	6.37	12.74
Pruned [12] + Fine-tuned	20/32	3.08 31.05	19.45 58.68	-154.69 30.03	5.97 52.43	10.14 47.03	-62.50 56.15	5.97	11.93
Pruned [16] + Fine-tuned	16/32	0.06 22.15	10.44 50.90	-182.25 -6.92	1.99 47.33	4.26 42.13	-124.75 42.51	5.16	10.32

Table 6: Comparison of iterative **pruning of 8 decoder layers** (which is the foundation of our pruning experiments) against **pruning 10, 12, and 16 decoder layers**. We observe that the quality of German degrades much more rapidly than that of Chinese, after pruning more than 8 layers. Moreover, pruning 16 layers from the Chinese model results in notably worse performance compared to pruning only 8, 10, or 12 layers, even after fine-tuning. When iteratively removing 16 layers from the German model, the removed layers are: [13, 3, 20, 9, 29, 1, 19, 27, 7, 26, 15, 18, 10, 17, 14, 30]. For the Chinese model, the removed layers are [7, 25, 3, 4, 29, 15, 26, 20, 24, 5, 10, 9, 27, 28, 18, 13]. The layers are listed from most to least important (left to right), according to the layer importance evaluation used in our iterative pruning method. In this experiment, all 32 encoder layers are kept intact; therefore, the table reports the remaining decoder layers as 24/32, 22/32, 20/32, and 16/32 encoder/decoder layers.

fine-tuning is performed 8 times, once after each pruning step.

By the end of the process, the pruned layers for German are [13, 1, 3, 17, 12, 4, 14, 21], based on layer importance evaluation. These layers differ from those selected when pruning without immediate recovery (cf. Section 4.3), since fine-tuning after each iteration changes the relative importance of the remaining layers.

Model	BLEU	chrF++	COMET
pruned + FT [after]	10.78 <b>32.16</b>	39.58 <b>60.39</b>	-44.40 <b>39.08</b>
pruned C FT [immediate]	30.96 31.67	59.02 59.76	30.07 36.13

Table 7: Comparison of fine-tuning after the end of the pruning process (+ FT [after]) against immediate fine-tuning after each pruning iteration (C FT [immediate]). Fine-tuning only after completing the pruning process achieves better final performance. The results are for iterative pruning of 8 decoder layers from the English-to-German model.

While this immediate recovery approach improved evaluation scores of the pruned model, it did not achieve performance gains over our standard method of fine-tuning only once after complete

pruning. This might be due to overfitting caused by fine-tuning several times on the small in-domain dataset. Moreover, immediate recovery through fine-tuning after each pruning iteration is much more computationally intensive.

#### 4.5 Pruning more layers

In our main pruning experiments (cf. Section 3.3), we pruned only 8 layers based on layer importance evaluation. We decided to experiment with pruning more layers to explore the level to which a model can be pruned while keeping a similar level of translation performance. As Table 6 illustrates, we compare pruning 8, 10, 12, and 16 layers. We observe that the quality after pruning up to 12 layers and fine-tuning the pruned model on the ACL 60/60 dataset is close to pruning 8 layers. However, when pruning 16 layers, the quality starts to degrade considerably. It is worth noting that the results reported in Table 6 are only for the pruning and initial fine-tuning steps, while Section 3.3 describes the whole process that involves knowledge distillation, extra compression with quantization, and further fine-tuning with QLoRA.

Pruning reduces storage footprint while accelerating inference speed by approximately 20% and 40% when 8 and 16 layers are pruned, respectively, compared to the unpruned baseline, after full-parameter fine-tuning of both models on the

English-to-German in-domain data. In contrast, 4-bit quantization as implemented in QLoRA reduces storage at the cost of inference speed. Given that the shared task prioritized minimizing storage requirements, we applied QLoRA to the pruned model in the final fine-tuning stage. For deployment scenarios where inference speed is critical, however, the pruned model can be fine-tuned with standard LoRA instead of QLoRA to avoid quantization overhead.

#### 4.6 Fine-tuning before/after pruning

It is common to fine-tune pruned models *after* pruning (Kim et al., 2023). As we pruned 1/4 of the decoder layers, obtaining valid translations for German was not possible without further training (cf. Table 2). Similarly, fine-tuning the teacher model on in-domain data *before* pruning is especially recommended for downstream tasks (Li et al., 2020).

#### 4.7 Out-of-Domain Data Size

After layer pruning, we added more data for fine-tuning the pruned model to recover the translation quality of the teacher model. We mixed the in-domain data (ACL 60/60), knowledge distillation data, and out-of-domain data from the CoVoST2 dataset. We experimented with different sizes of out-of-domain data, namely 10k, 50k, 80k and 100k randomly sampled segments.

Data Size	BLEU	chrF++	COMET
10k	37.83	63.84	48.90
50k	38.47	64.09	54.56
80k	<b>40.24</b>	65.04	54.75
100k	39.59	<b>65.14</b>	<b>59.21</b>

Table 8: Investigating the effect of the data sizes used from the out-of-domain CoVoST2 dataset. Increasing out-of-domain data from 10k to 50k or 80k improves the translation quality on the English-to-German test split of the ACL 60/60 dataset. When increasing the out-of-domain data from 80k to 100k, this only improves the COMET score, but not the BLEU and chrF++ scores.

As Table 8 shows, for English-to-German translation, increasing the data size from 10k to 50k and 80k segments improves the overall performance. However, increasing the data size to 100k has diminishing returns, especially in terms of BLEU and chrF++ scores, while it improves the COMET score.

## 5 Inference and Evaluation

For inference, we use greedy generation by disabling the sampling options. We apply the prompt illustrated in Section 4.1. We use a batch size of 1, and set the generation max length to 1024 tokens.

To evaluate our systems, we calculated BLEU (Papineni et al., 2002), chrF++ (Popović, 2017), as implemented in the sacreBLEU library<sup>7</sup> (Post, 2018). While we use chrF++ for German, we use raw chrF for Chinese, following the author of the metric who noted that “the concept of *Chinese words* is generally not clear” (Popović, 2017). For semantic evaluation, we use COMET (Rei et al., 2020).<sup>8</sup> Table 1 and Table 2 report the results of the main experiments. Moreover, we conduct an ablation study (cf. Section 4) to investigate the effect of modifying some aspects of our experiments, such as the baseline model, the pruning approach, the number of pruned layers, and the data size.

## 6 Conclusions and Future Work

In this work, we showed that combining multiple compression techniques enables substantial model size reduction with minimal impact on speech translation performance. To conclude, QLoRA fine-tuning with knowledge distillation achieved superior translation quality compared to layer pruning alone, though with reduced model compression. To achieve higher compression ratios while preserving translation quality, we employed a combined approach using iterative layer pruning, quantization, knowledge distillation, and multi-stage fine-tuning. The code of our experiments is publicly available.<sup>9</sup>

In future work, we plan to explore adaptive compression strategies that dynamically adjust pruning levels and quantization precision based on real-time deployment constraints such as memory limits and latency requirements. Additionally, we aim to evaluate our compression techniques across more diverse datasets, including both authentic and synthetic training data, to better understand the generalization capabilities of our approach. Given that Qwen2-Audio-Instruct relies on text prompts for generation, it would be interesting to investigate retrieval-augmented generation with few-shot prompting to improve the translation quality of compressed models.

<sup>7</sup><https://github.com/mjpost/sacrebleu>

<sup>8</sup>In particular, we used the “wmt20-comet-da” model.

<sup>9</sup><https://github.com/ymoslem/Model-Compression>



## References

- Idris Abdulmumin, Victor Agostinelli, Tanel Alumäe, Antonios Anastasopoulos, Ashwin, Luisa Bentivogli, Ondřej Bojar, Claudia Borg, Fethi Bougares, Roldano Cattoni, Mauro Cettolo, Lizhong Chen, William Chen, Raj Dabre, Yannick Estève, Marcello Federico, Marco Gaido, Dávid Javorský, Marek Kasztelnik, Tsz Kin Lam, Danni Liu, Evgeny Matusov, Chandresh Kumar Maurya, John P McCrae, Salima Mdhaffar, Yasmin Moslem, Kenton Murray, Satoshi Nakamura, Matteo Negri, Jan Niehues, Atul Kr Ojha, John E Ortega, Sara Papi, Pavel Pecina, Peter Polák, Piotr Połec, Beatrice Savoldi, Nivedita Sethiya, Claytone Sikasote, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Brian Thompson, Marco Turchi, Alex Waibel, Patrick Wilken, Rodolfo Zevallos, Vilém Zouhar, and Maike Züfle. 2025. Findings of the IWSLT 2025 Evaluation Campaign. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, Vienna, Austria (in-person and online). Association for Computational Linguistics.
- Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. 2024. *Qwen2-Audio Technical Report*. *arXiv [eess.AS]*.
- Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. *Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models*. *arXiv [eess.AS]*.
- Josep Crego and Jean Senellart. 2016. *Neural Machine Translation from Simplified Translations*. *arXiv [cs.CL]*.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. *QLoRA: Efficient Finetuning of Quantized LLMs*. *arXiv [cs.LG]*.
- Sanchit Gandhi, Patrick von Platen, and Alexander M Rush. 2023. *Distil-Whisper: Robust knowledge distillation via large-scale pseudo labelling*. *arXiv [cs.CL]*.
- 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*. *arXiv [cs.CL]*.
- Wandri Jooste, Andy Way, Rejwanul Haque, and Riccardo Superbo. 2022. *Knowledge Distillation for Sustainable Neural Machine Translation*. In *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Volume 2: Users and Providers Track and Government Track)*, pages 221–230, Orlando, USA. Association for Machine Translation in the Americas.
- Damjan Kalajdzievski. 2023. *A rank stabilization scaling factor for fine-tuning with LoRA*. *arXiv [cs.CL]*.
- Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. 2023. *SOLAR 10.7B: Scaling large language models with simple yet effective depth up-scaling*. *arXiv [cs.CL]*.
- 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, Austin, Texas. Association for Computational Linguistics.
- Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. *Datasets: A community library for natural language processing*. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, K Keutzer, D Klein, and Joseph E Gonzalez. 2020. *Train large, then compress: Rethinking model size for efficient training and inference of transformers*. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, pages 5958–5968, Virtual.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. *Bleu: a Method for Automatic Evaluation of Machine Translation*. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- David Peer, Sebastian Stabinger, Stefan Engl, and Antonio Rodríguez-Sánchez. 2022. *Greedy-layer pruning: Speeding up transformer models for natural language processing*. *Pattern Recognit. Lett.*, 157:76–82.
- Yifan Peng, Jaesong Lee, and Shinji Watanabe. 2023a. *I3D: Transformer architectures with input-dependent dynamic depth for speech recognition*. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.
- Yifan Peng, Yui Sudo, Shakeel Muhammad, and Shinji Watanabe. 2023b. *DPHuBERT: Joint Distillation and Pruning of Self-Supervised Speech Models*. In *Proceedings of the 24th Annual Conference of the*

- International Speech Communication Association, *InterSpeech 2023*, pages 62–66.
- Maja Popović. 2017. [chrF++: words helping character n-grams](#). In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.
- 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, Brussels, Belgium. Association for Computational Linguistics.
- Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. [COMET: A Neural Framework for MT Evaluation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.
- Pedram Rostami and Mohammad Javad Dousti. 2024. [CULL-MT: Compression using language and layer pruning for machine translation](#). *arXiv [cs.CL]*.
- Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2023. [On the effect of dropping layers of pre-trained transformer models](#). *Comput. Speech Lang.*, 77(101429):101429.
- Elizabeth Salesky, Kareem Darwish, Mohamed Al-Badrashiny, Mona Diab, and Jan Niehues. 2023. [Evaluating multilingual speech translation under realistic conditions with resegmentation and terminology](#). In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 62–78, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Marcos Treviso, Ji-Ung Lee, Tianchu Ji, Betty van Aken, Qingqing Cao, Manuel R Ciosici, Michael Hassid, Kenneth Heafield, Sara Hooker, Colin Raffel, Pedro H Martins, André F T Martins, Jessica Zosa Forde, Peter Milder, Edwin Simpson, Noam Slonim, Jesse Dodge, Emma Strubell, Niranjana Balasubramanian, Leon Derczynski, Iryna Gurevych, and Roy Schwartz. 2023. [Efficient methods for natural language processing: A survey](#). *Transactions of the Association for Computational Linguistics*, 11:826–860.
- 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 (NIPS 2017)*, volume 30. Curran Associates, Inc.
- Changhan Wang, Anne Wu, and Juan Pino. 2021. [CoVoST 2 and Massively Multilingual Speech-to-Text Translation](#). In *Proceedings of the 22nd Annual Conference of the International Speech Communication Association, InterSpeech 2021*, pages 2247–2251.
- Haoyu Wang, Siyuan Wang, Wei-Qiang Zhang, Suo Hongbin, and Yulong Wan. 2023. [Task-Agnostic Structured Pruning of Speech Representation Models](#). In *Proceedings of the 24th Annual Conference of the International Speech Communication Association, InterSpeech 2023*, pages 231–235.
- Haryo Akbarianto Wibowo, Haiyue Song, Hideki Tanaka, Masao Utiyama, Alham Fikri Aji, and Raj Dabre. 2025. [IteRABRe: Iterative recovery-aided block reduction](#). *arXiv [cs.CL]*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Stroudsburg, PA, USA. Association for Computational Linguistics.