DIET-KIT : Post-Training Quantization for Speech LLMs

Danni Liu **Sai Koneru** **Jan Niehues**
Karlsruhe Institute of Technology, Germany
{danni.liu, sai.koneru, jan.niehues}@kit.edu

Abstract

We present Diet-KIT, a system for the IWSLT speech translation compression task under a strict 4 GB on-disk storage constraint, starting from the 16 GB Qwen2-Audio-7B base model. Compression is achieved with a sequential pipeline based on Half-Quadratic Quantization (HQQ). Based on systematic ablations, we find that 4-bit quantization preserves translation quality well, whereas 3-bit quantization induces a sharp performance cliff, precluding aggressive compression across the whole model. We further show that the embedding table tolerates 2-bit quantization with negligible loss, while the LM head requires higher precision. To satisfy the storage constraint, we propose a sensitivity-guided layer selection method that identifies MLP sublayers tolerant to 3-bit compression via a per-layer sensitivity analysis, which consistently outperforms manual and random layer selection. Finally, Activation-aware Weight Quantization (AWQ) calibration is applied as a data-driven refinement stage. The final system achieves 3.98 GB on disk with COMET scores of 74.4 on en→de and 77.1 on en→zh, compared to 75.6 and 79.5 for the uncompressed fine-tuned model.

1 Introduction

Large language models that support speech inputs have advanced rapidly in recent years, with models such as Qwen2-Audio (Chu et al., 2024) demonstrating strong performance across a range of spoken language tasks. However, the deployment of such models in real-world settings remains challenging. State-of-the-art speech LLMs typically require tens of gigabytes of storage and significant GPU memory, limiting their accessibility in resource-constrained environments.

Model compression has emerged as a practical solution to this problem, with post-training quantization in particular offering possibilities to reduce model size without the cost of retraining. While quantization has been widely studied for text-only LLMs (Dettmers et al., 2022; Frantar and Alistarh, 2023; Lin et al., 2024; Badri and Shaji, 2023), its behavior in speech models remains less explored.

In this work, we investigate the compression of Qwen2-Audio-7B for speech translation as part of the IWSLT 2026 shared task on compression (Adelani et al., 2026), which imposes a 4 GB on-disk storage constraint on a model that originally occupies 16 GB. We develop Diet-KIT, a sequential compression pipeline combining HQQ quantization, embedding optimization, sensitivity-guided layer selection, and AWQ calibration, and provide a detailed ablation of each component. Our main findings are:

- 4-bit linear quantization preserves translation quality with minimal degradation while 3-bit induces a sharp performance cliff.
- The input embedding table is surprisingly robust to 2-bit quantization, whereas the LM head is highly sensitive and requires higher precision.
- Uniform quantization strategies are inefficient at the compression frontier, and a lightweight per-layer sensitivity analysis based on cross-entropy reliably identifies the MLP sublayers most resilient to aggressive compression.
- Contrary to findings in text-only settings, the size of AWQ calibration set has no consistent effect on downstream quality, suggesting that activation statistics in speech LLMs are less stable and warrant further studies.

2 Related Work

The primary objective of model compression is to reduce the model size while preserving as much of the pretrained knowledge as possible in a compact representation. This knowledge transfer is typically achieved in two broad paradigms:

Distillation: This process generally involves two stages. First, a smaller *student* model is derived

Proceedings of the 23rd International Conference on Spoken Language Translation (IWSLT 2026), pages 189–196

#	Model	Size (GB)	Compression Rate (%)	EN→DE		EN→ZH	
				BLEU	COMET	BLEU	COMET
1	Qwen2-Audio-7B	16.80	–	15.0	61.3	23.1	67.0
2	Qwen2-Audio-7B (fine-tuned)	16.80	–	27.0	75.6	41.0	79.5
3	*(2)* + HQQ 8-bit	8.86	47.3	26.8	75.6	40.9	79.3
4	*(2)* + HQQ 4-bit	5.26	68.7	24.4	74.0	40.5	78.1
5	*(2)* + HQQ 3-bit	4.53	73.0	11.0	63.8	15.6	65.8
6	*(4)* + embedding 4-bit	4.41	73.8	24.9	74.2	39.8	78.2
7	*(4)* + embedding 2-bit	4.26	74.6	24.2	73.6	38.5	78.1
8	*(7)* + group size 64→128	4.03	76.0	25.3	73.9	37.4	77.8
9	*(7)* + group size 64→256	3.92	76.7	22.1	72.7	33.3	76.6
10	*(7)* + group size 64→512	3.86	77.0	19.5	70.0	33.0	75.3
11	*(8)* + layer-selective 3-bit MLP (Table 4)	3.98	76.3	24.8	73.7	36.4	77.6
12	*(11)* + AWQ calibration (Table 5)	3.98	76.3	24.3	74.4	35.2	77.1

Table 1: Translation performance (BLEU / COMET) and compression statistics for Qwen2-Audio-7B variants on ACL 60/60 test split (Salesky et al., 2023) with gold segmentation. System 12 is our final system, while system 11 is the contrastive system. Size cell shading reflects proximity to the 4 GB storage threshold.

Sample ID	Qwen2 Audio 7B Output	Reference Translation	Failure Type
1	[Musik] Hallo, alle. Heute werde ich unser Forschungsprojekt vorstellen: Lernen zu reason deduktiv, mathematische Problemlösung als komplexe logische Abstraktion.	Hallo zusammen. Heute werde ich unsere Forschungsarbeit Learning to Reason Deductively: Math Word Problem Solving as Complex Relation Extraction vorstellen.	Tag
2	[Musik]	Ich bin Allan vom ByteDance AI Lab, und dies ist eine gemeinsame Arbeit mit Jierui Li von der University of Texas in Austin und Wei Lu von SUTD.	Tag
117	[Q] Zuletzt die Satzartikel sind ein kleines Paragraph, das normalerweise die typische Entnahmeeinheit in 大多数 Entnahmearbeiten ist.	Schließlich handelt es sich bei den Gesetzesartikeln nicht um kleine Absätze, die in den meisten Retrieval-Arbeiten die typische Retrieval-Einheit darstellen.	Tag, Wrong Language
303	Fünf Methoden wurden verglichen: C, C, C, C, C, C, C, C, C, C, C, C, C, C, C, C, ...	Fünf Methoden wurden verglichen: CEAS, CAS-Single, CAS-Multi, Clustering und Glyph aus einer früheren Studie.	Repetition/ Hallucination

Table 2: Examples illustrating failure cases when translating with Qwen2 Audio (row 1 of Table 1). The audio sample ID indicates the segment number in the ACL 60/60 test split (gold segmentation).

from a larger *teacher* model, often via structured (Moslem, 2025) or unstructured pruning (Kim et al., 2024; Chen et al., 2025) or randomly initialized (Hinton et al., 2015). Subsequently, the student model is trained to mimic the teacher by leveraging additional supervision signals, such as token-level probability distributions (Wang et al., 2020; Wei et al., 2024), hidden representations (Chen et al., 2025), or synthetic data generated by the teacher (Kim and Rush, 2016; Finkelstein et al., 2024). This training procedure helps constrain the student model, preventing it from deviating significantly from the teacher and thereby preserving the pre-trained knowledge.

Quantization: Unlike distillation, quantization requires little to no retraining. The idea is to reduce memory usage by converting model parameters to lower-precision numerical formats, ideally with minimal performance degradation. Furthermore, it remains unclear which parameters contribute most to this degradation, particularly in the context of speech translation.

3 Adapting Qwen2-Audio for Speech Translation

Our primary objective is to study the behavior of quantization in general-purpose speech LLMs. Ideally, this analysis would be conducted on the pre-trained model directly, keeping the system within the constrained track of the shared task. However, as shown in row 1 of Table 1, the pretrained Qwen2-Audio-7B model[1] (Chu et al., 2024) required in the shared task exhibits poor translation performance on both translation directions, with representative failure cases illustrated in Table 2. Quantizing a model that cannot perform the target task would conflate quantization-induced degradation with the pre-existing lack of translation capability, making it difficult to draw conclusions about compression behavior. Therefore, a sufficiently capable starting

[1] Qwen/Qwen2-Audio-7B

point is a prerequisite for our subsequent quantization steps.

In this light, we fine-tune Qwen2-Audio-7B on speech translation data sampled from CoVoST (Wang et al., 2021) and Europarl-ST (Iranzo-Sánchez et al., 2020), which renders our submission unconstrained.[2] While training on the full available data could further improve performance, doing so is not the goal here. We instead sample a controlled multilingual subset spanning several language pairs beyond those evaluated in the shared task, as detailed in Table 3, to obtain a model that is competent at translation without being narrowly optimized for the evaluation directions.

Language	# Utt.	# Hrs	Language	# Utt.	# Hrs
German	49997	87.38	French	15000	35.67
Turkish	34999	52.10	Dutch	15000	35.47
Chinese	34999	52.05	Italian	15000	37.79
Japanese	34996	51.97	Polish	15000	35.39
Spanish	15000	35.88	Portuguese	15000	35.51
Romanian	15000	35.34			

Table 3: Number of utterances and hours of training data sampled per language in the combined CoVoST and Europarl-ST dataset.

For adaptation, we employ LoRA-based fine-tuning rather than full fine-tuning.[3] This choice is motivated by two factors: (i) full fine-tuning tends to cause larger deviations from the pretrained model, which is undesirable in our analysis setting, and (ii) it incurs significantly higher computational cost. The hyperparameters used for fine-tuning are reported in Table 7 in Appendix A.

4 Quantization

We approach compression as a sequential pipeline, where each stage builds on the previous one and is evaluated against the 4 GB storage constraint imposed in the shared task. We begin by selecting a quantization method and establishing baselines (§4.1), then sequentially refine the configuration by embedding optimization (§4.2), group size tuning (§4.3), layer-selective compression (§4.4), and finally data-driven calibration (§4.5).

[2]The constrained track only allows ACL 60/60 data. Use of additional training data makes our system unconstrained.

[3]During LoRA training, additional parameters are introduced, but at inference time the adapters are merged back into the base model, leaving the total parameter count unchanged.

4.1 Base Quantization with HQQ

Why HQQ? A requirement by the task organizers is reducing the on-disk model size to 4 GB or below, starting from Qwen2-Audio-7B which has 16 GB at half precision (16 bit). This makes the choice of serialization approach as important as the quantization algorithm itself. The standard quantization backend in HuggingFace Transformers is bitsandbytes (BNB) (Dettmers et al., 2022), which offers 4-bit quantization via NF4, a non-uniform grid optimized for normally-distributed weights, but performs *no optimization* of the quantization parameters themselves (zero-points and scales are set without minimizing weight reconstruction error). More crucially for our purposes, BNB compresses weights *only at runtime*, while serializing *original-precision* weights to disk. The on-disk checkpoint therefore reflects the original model size regardless of quantization, making BNB less suitable in this case[4]. We instead apply Half-Quadratic Quantization (HQQ) (Badri and Shaji, 2023), which formulates quantization as a weight-error minimization problem optimized via a Half-Quadratic solver. Unlike round-to-nearest approaches, this explicitly minimizes the discrepancy between original and quantized weights, making it substantially more accurate at 4-bit precision where unoptimized quantization suffers from large representation errors on outlier weights (Dettmers et al., 2022). Our pipeline applies HQQ quantization prior to serialization, saving the resulting state_dict via safetensors. This captures the packed integer tensors and associated quantization metadata (W_q, scales, zero-points) directly, ensuring the on-disk checkpoint satisfies the storage constraint.

Quantization Cliff from 4-bit to 3-bit As shown in Table 1 (rows 3 and 4), quantization to 8-bit and 4-bit preserves translation quality well: the 8-bit model suffers negligible degradation across both language pairs, and even at 4-bit the drop in COMET is limited to 1.6 points of the fine-tuned baseline on en→de (74.0 vs. 75.6) and 1.4 points on en→zh (78.1 vs. 79.5). However, compressing further to 3-bit reveals a sharp performance cliff. BLEU drops from 24.4 to 11.0 on en→de and from 40.5 to 15.6 on en→zhm, while COMET falls by over 10 points on both directions, collapsing to levels comparable to the untuned model in row 1. This suggests that at 3-bit, HQQ can no longer

[4]We note that newer BNB versions introduced 4-bit serialization support but requires special load-side handling.

represent the fine-tuned weight distribution, and the model loses the translation behavior acquired during fine-tuning. We therefore treat this 4-bit model as the basis of subsequent efforts.

4.2 Embedding Optimization

By default, HQQ quantizes only linear layers. However, since the embedding table and output projection (LM head) constitute a substantial share of total parameters, we explore targeted optimization of these components.

Embedding Table Robust to Quantization We apply HQQ quantization to the input embedding table at both 4-bit and 2-bit, building on top of the 4-bit model from the previous section. The results show that the embedding is comparatively robust to aggressive quantization: compressing it to 4-bit (row 6) reduces the model size to 4.41 GB with negligible impact on translation quality (COMET: 74.2 en→de, 78.2 en→zh), and even 2-bit embedding quantization (row 7, 4.26 GB) causes only marginal further degradation, with COMET scores within 0.4–0.5 points of the 4-bit linear-only baseline.

LM Head Sensitive to Quantization Applying the same aggressive 2-bit quantization to the LM head shows a different picture: It causes a complete loss of generation capability. A potential reason is that, as the LM head projects the final hidden states onto the full vocabulary, its weight distribution is tightly coupled to the model's output calibration. We therefore leave the LM head in the previous 4-bit precision in subsequent configurations.

Limitations of Vocabulary Pruning An alternative approach for reducing embedding size is vocabulary pruning, i.e., removing tokens that are not used by the target languages. However, an analysis of Qwen2's vocabulary reveals that this provides negligible practical gain for our en→{de,zh} setting. Of the 151,643 tokens in the vocabulary of Qwen2, the vast majority cannot be safely removed. Latin-script tokens account for 62.0% and are required for German, CJK (Chinese-Japanese-Korean) tokens account for 17.1% and are required for Chinese, and punctuation and digit tokens (12.5%) are required for both directions. Only script-specific tokens in e.g., Cyrillic, Arabic are safely prunable, but they only amount to approximately around 8% of the vocabulary. At the current compression level, this translates to roughly 20 MB of storage saving, which is marginal relative to the overall compression budget.

4.3 HQQ Group Size Optimization

In HQQ, quantization parameters, i.e., scales and zero-points, are computed independently for contiguous blocks of weights, or *groups*. A smaller group size means more quantization parameters are stored alongside the packed weights, which improves fidelity at the cost of additional metadata overhead. The default group size in HQQ is 64 in the official implementation[5]. We explore coarser groupings of 128, 256, and 512, applied on top of the 2-bit embedding configuration from the previous section.

The results exhibits a clear trade-off. Increasing the group size from 64 to 128 (row 8) reduces the model size to 4.03 GB with only modest quality loss: COMET drops by 0.7 points on en→de and 0.3 points on en→zh relative to row 7. However, the size of 4.03 GB still slightly exceeds the 4 GB storage constraint, and the remaining gap cannot be closed by group size alone without substantial quality loss. Pushing further to group sizes of 256 and 512, despite fitting within the 4GB limit, results in increasingly steep quality degradation, with BLEU on en→de falling from 25.3 to 22.1 and 19.5, and COMET declining by over 4 points at group size 512 relative to group size 128. Based on this, we adopt group size 128, and explore more measures to close the remaining size gap without relying on coarser groupings that sacrifice too much translation performance.

4.4 Layer-Selective Quantization

As shown in row 5 of Table 1, applying 3-bit quantization to all linear layers causes detrimental quality loss. We hypothesize that MLP layers are more robust to aggressive quantization than attention layers, whose weights encode token-interaction patterns that are potentially more sensitive to quantization errors. To understand this, we first quantize only the MLP layers globally to 3-bit while retaining attention layers at 4-bit. Although this configuration still degrades translation quality, the loss is more moderate compared to the drastic degradation under uniform 3-bit quantization. This motivates us towards a more *selective* strategy.

Sensitivity-Guided Layer Selection Rather than quantizing all MLP layers uniformly, we seek to

[5] https://github.com/dropbox/hqq

identify the minimal subset whose 3-bit compression is sufficient to bring the group size 128 model (row 8, 4.03 GB) below the 4 GB storage limit, while minimizing quality impact. To determine which layers are most tolerant to compression, we conduct a sensitivity analysis. Starting from the row 8 checkpoint, we individually swap the MLP of each Transformer layer to 3-bit and measure the resulting change in teacher-forced cross-entropy loss on the ACL 60/60 development set. The direct loss calculation is substantially faster than running full inference and computing translation scores for each candidate configuration as used by Moslem (2025), making it practical to analyze the full model depth at fine granularity. Layers are then ranked by their induced loss deviation, and the top n least sensitive layers, where n is the smallest number sufficient to push the model size below 4 GB, are selected for 3-bit MLP quantization in the final configuration. In our case, at least 4 MLP layers need to be in 3 bit to reach the storage threshold, reducing the model from 4.03 GB to 3.98 GB.

Configuration	EN→DE		EN→ZH	
	BLEU	COMET	BLEU	COMET
Row 8 Table 1	25.3	73.9	37.4	77.8
+ Selected layers	**24.8**	**73.7**	**36.4**	**77.6**
+ First 4 layers	20.8	71.5	30.4	75.2
+ Last 4 layers	19.7	72.1	29.5	75.5
+ Middle 4 layers	24.7	73.5	32.7	76.5
+ Random 4 layers	22.1	72.4	30.7	75.9

Table 4: Comparison of MLP layer selection strategies for 3-bit quantization of 4 layers to push the model size under 4 GB (4.03→3.98 GB), built on top of row 8 of Table 1. "Selected" refers to the least sensitive layers identified via cross-entropy sensitivity analysis (layers 4, 21, 22, 24).

Table 4 compares the sensitivity-guided selection against manual and random selection. The selected layers consistently outperform all alternatives across both language pairs. Manual selection perform poorly and inconsistently: the first and last 4 layers incur the largest quality drops. The middle 4 layers are competitive on en→de but degrade substantially on en→zh, illustrating that positional heuristics do not generalize reliably across language pairs. To further validate that the gains of the proposed selection are not coincidental, we compare against randomly selected sets of 4 layers averaged over 3 independent runs. The random baseline also falls well short of the sensitivity-guided selection, confirming that the performance advantage is attributable to the analysis.

4.5 AWQ Calibration

All compression steps so far operate only on the model weights *without* reference to any data. While HQQ minimizes weight reconstruction error, it has no knowledge of which weights matter most for the model's actual output behavior. Activation-Aware Weight Quantization (AWQ) (Lin et al., 2024) addresses this by incorporating a small calibration set. AWQ is motivated by the observation that not all weights contribute equally to model performance. Rather than using weight magnitudes alone, it identifies salient weight channels via the distribution of input activations, and applies a per-channel scaling transformation to protect them.

Calibration Data	Size	EN→DE		EN→ZH	
		BLEU	COMET	BLEU	COMET
None (row 10, Table 1)	–	24.8	73.7	36.4	**77.6**
Source only	128	24.8	74.1	35.4	77.4
	256	24.0	73.9	35.3	**77.6**
	All	24.3	**74.4**	35.2	77.1
EN→DE pairs	128	25.0	74.0	37.7	**77.7**
	256	24.4	73.8	35.5	77.2
	All	24.3	74.0	35.8	77.3
EN→ZH pairs	128	24.8	73.9	36.5	77.3
	256	24.6	73.7	36.6	**77.7**
	All	24.5	73.7	37.1	77.5

Table 5: Effect of AWQ calibration set size and composition on translation quality, applied on top of the final compressed model (row 10 of Table 1).

Table 5 presents the results of across calibration data compositions (source audio only, or using audio-translation pairs) and sizes (128, 256, or full dev set of 468 samples), with COMET as the primary metric. The effect of calibration set size is inconsistent across conditions. Unlike findings from existing calibration studies on text-only language models, which generally report that a small calibration set of 128 samples is sufficient (Frantar and Alistarh, 2023; Lin et al., 2024; Williams and Aletras, 2024), there is no clear trend here. The best COMET scores distributed across all three size settings depend on calibration data condition and language. This may reflect the additional complexity of a speech LLM compared to pure text settings.

For calibration data composition, paired translation data does not consistently outperform source-only input despite additional target-side context.

System	ACL	TVSeries	ChallengeAccent	CallCenter	YouTube	BusinessNews
Contrastive (#11)	75.6	**49.0**	36.4	**62.7**	**51.9**	61.6
Primary (#12)	**76.7**	47.4	36.3	61.7	50.4	**63.0**

Table 6: COMET scores on the blind test sets for the primary (system 12) and contrastive (system 11) submissions. Official test sets are segmented with SHAS (Tsiamas et al., 2022); the best score per test set is shown in bold.

Therefore, we select the source-only calibration with the full development set as our final configuration, as it yields the strongest COMET of 74.4 on en→de, the direction more challenging for Qwen2-Audio given its strong pre-existing Chinese capabilities.

5 Final System

From Table 1, we submitted system 12 as our primary submission and system 11 as the contrastive system. While the scores reported throughout this paper are computed on the ACL 60/60 test set with gold segmentation, the official test sets are unsegmented[6]. We therefore apply SHAS (Tsiamas et al., 2022) with default parameters for segmentation. Manual inspection reveals that a number of segments are dropped on noisier test sets, likely due to SHAS's sensitivity to audio quality. Replacing SHAS with a more robust segmentation approach could potentially resolve this, but as this is orthogonal to the quantization focus of this work, we leave it as a direction for future work.

Results on Blind Test Set The results on the blind test sets are shown in Table 6. The AWQ-calibrated final system improves performance on the in-domain data, but these gains do not transfer consistently to the out-of-domain test sets. Moreover, the low scores on TVSeries, ChallengeAccent, and YouTube indicate that the systems are of limited practical use on these domains, owing to repetitions and blank outputs. While we expect that a length penalty together with a more robust segmenter would partly mitigate these issues, the behavior nonetheless highlights a broader robustness concern for multimodal LLMs.

6 Conclusion

We presented Diet-KIT, a compression pipeline for Qwen2-Audio-7B for the IWSLT 2025 speech translation shared task, reducing the model from 16 GB to 3.98 GB while retaining competitive translation quality. Our analysis demonstrates that 4-bit HQQ quantization constitutes a robust compression base for speech LLMs, beyond which uniform strategies cause catastrophic degradation. Position-specific measures, including 2-bit embedding quantization, sensitivity-guided 3-bit MLP layer selection allow the storage constraint to be satisfied with minimal additional quality loss. We hope that our findings provide a useful reference for future works on efficient speech LLM deployment.

6.1 Future Directions

While the results presented in this work primarily focus on the effects of quantization, there are several promising directions for further improvement.

First, incorporating layer drop (Fan et al., 2020) for model compression is non-trivial. Effectively integrating layer drop during training to obtain a pruned model is challenging, as such training is often unstable and typically requires carefully designed scheduling strategies (Zhang and He, 2020). Our preliminary experiments also showed below par performances when pruning model around 50% of layers. Exploring robust and effective layer drop schemes for ST models remains an important gap for future research.

Building on this, combining layer drop with layer-wise distillation presents a compelling approach (Chen et al., 2025). In this setup, intermediate layers of the compressed model can be trained to match the representations of the corresponding layers in the original model, thereby providing stronger supervision and facilitating better knowledge transfer.

Furthermore, this framework could be extended with early-exit mechanisms (Elhoushi et al., 2024), which dynamically adjust the number of layers used during inference on a per-example basis or quantize specific layers (Yang et al., 2025). Such approaches have the potential to significantly improve computational efficiency while maintaining performance.

[6] except the accent test set

Acknowledgments

We thank the reviewers for helpful feedback. The authors gratefully acknowledge the computing time provided on the high-performance computer HoreKa by the National High-Performance Computing Center at KIT (NHR@KIT). This center is jointly supported by the Federal Ministry of Education and Research and the Ministry of Science, Research and the Arts of Baden-Württemberg, as part of the National High-Performance Computing (NHR) joint funding program (https://www.nhr-verein.de/en/our-partners). HoreKa is partly funded by the German Research Foundation (DFG). Part of this work was funded by the KiKIT (The Pilot Program for Core-Informatics at the KIT) of the Helmholtz Association. This work has received funding from the European Union's Horizon Europe Framework under grant agreement No 101213369, project DVPS (Diversibus Viis Plurima Solvo).

References

David Ifeoluwa Adelani, Victor Agostinelli, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Sebastien Bratières, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, Lizhong Chen, Marcello Federico, Marco Gaido, Mahendra Gupta, HyoJung Han, Ali Hatami, David Javorský, Yejin Jeon, Marek Kasztelnik, Antoine Laurent, and 33 others. 2026. Speech translation and metrics in 2026: Findings of the iwslt campaign. In *Proceedings of the 23rd International Conference on Spoken Language Translation (IWSLT 2026)*, San Diego, California, US. Association for Computational Linguistics.

Hicham Badri and Appu Shaji. 2023. Half-quadratic quantization of large machine learning models.

Xiaodong Chen, Yuxuan Hu, Jing Zhang, Yanling Wang, Cuiping Li, and Hong Chen. 2025. Streamlining redundant layers to compress large language models. In *The Thirteenth International Conference on Learning Representations*.

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 preprint arXiv:2407.10759*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pages 30318–30332.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, and 1 others. 2024. LayerSkip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12622–12642. Association for Computational Linguistics.

Angela Fan, Edouard Grave, and Armand Joulin. 2020. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations*.

Mara Finkelstein, David Vilar, and Markus Freitag. 2024. Introducing the NewsPaLM MBR and QE dataset: LLM-generated high-quality parallel data outperforms traditional web-crawled data. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1355–1372. Association for Computational Linguistics.

Elias Frantar and Dan Alistarh. 2023. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020. Europarl-ST: A multilingual corpus for speech translation of parliamentary debates. In *ICASSP 2020 – IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8229–8233.

Bo-Kyeong Kim, Geonmin Kim, Tae-Ho Kim, Thibault Castells, Shinkook Choi, Junho Shin, and Hyoung-Kyu Song. 2024. Shortened LLaMA: Depth pruning for large language models with comparison of retraining methods. *arXiv preprint arXiv:2402.02834*.

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.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems 6 (MLSys 2024)*. mlsys.org.

Yasmin Moslem. 2025. Efficient speech translation through model compression and knowledge distillation. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 379–388. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. 2022. SHAS: Approaching optimal segmentation for end-to-end speech translation. In *Interspeech 2022*, pages 106–110. ISCA.

Changhan Wang, Anne Wu, Jiatao Gu, and Juan Pino. 2021. CoVoST 2 and massively multilingual speech translation. In *Interspeech 2021*, pages 2247–2251.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Fei Huang, and Kewei Tu. 2020. Structure-level knowledge distillation for multilingual sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3317–3330. Association for Computational Linguistics.

Jingxuan Wei, Linzhuang Sun, Yichong Leng, Xu Tan, Bihui Yu, and Ruifeng Guo. 2024. Sentence-level or token-level? A comprehensive study on knowledge distillation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 6531–6540.

Miles Williams and Nikolaos Aletras. 2024. On the impact of calibration data in post-training quantization and pruning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10100–10118. Association for Computational Linguistics.

Ning Yang, Fangxin Liu, Junjie Wang, Tao Yang, Kan Liu, Haibing Guan, and Li Jiang. 2025. DASH: Input-aware dynamic layer skipping for efficient LLM inference with markov decision policies. *arXiv preprint arXiv:2505.17420*.

Minjia Zhang and Yuxiong He. 2020. Accelerating training of transformer-based language models with progressive layer dropping. In *Advances in Neural Information Processing Systems*, volume 33, pages 14011–14023.

A Appendix

Hyperparameter	**Value**
Model ID	Qwen/Qwen2-Audio-7B
Number of Epochs	3.0
Train Batch Size (per device)	8
Eval Batch Size (per device)	8
Gradient Accumulation Steps	1
Learning Rate	1×10^{-6}
Warmup Ratio	0.03
Logging Steps	10
Save Steps	2000
Evaluation Steps	2000
bf16	True
Gradient Checkpointing	True
Seed	42
LoRA Rank (r)	64
LoRA Alpha	128
LoRA Dropout	0.05
Prompt Template	`<|audio_bos|><|AUDIO|><|audio_eos|> Translate this speech to {target_lang}:\n`

Table 7: Training hyperparameters for fine-tuning Qwen2-Audio-7B. All other parameters are set to their default values.