Selected-Layer Codec Compression for Compact Speech Translation Models: An IWSLT 2026 English-to-Chinese Submission

Alonso Palomino
mail@<first>pg.com

Abstract

This paper describes a selected-layer codec compression approach submitted to the IWSLT 2026 Model Compression Shared Task for constrained English-to-Chinese speech translation. The approach is compared against standard quantization, global codec compression, and a pruning-plus-codec variant. The results indicate that translation quality after compression depends strongly on where compression is applied. In these experiments, selected-layer compression preserves translation quality better than uniform global compression, with one variant achieving the highest COMET score among compressed systems and another providing the strongest overall quality-compression trade-off among the custom codec methods. These results suggest that simple layer-aware post-hoc compression is a viable approach for model compression in constrained English-to-Chinese speech translation.

1 Introduction

Recent progress in machine translation and speech translation has been driven by increasingly large neural models, but these gains come with substantial costs in memory footprint, storage, and inference latency, creating a practical tension between model quality and deployability, especially in realistic settings with hardware, energy, or bandwidth constraints. In multilingual machine translation, compression can preserve strong average performance while still degrading unevenly across languages and conditions (Mohammadshahi et al., 2022), and in speech translation the challenge is even sharper because long audio inputs, multimodal processing, and autoregressive generation further increase computational cost (Salesky et al., 2023; Gaido et al., 2024). This problem has become more pressing with the rise of audio-language models and large language model based speech translation systems, which can achieve strong end-to-end performance but remain difficult to deploy efficiently without compression or adaptation (Gaido et al., 2024; Chen et al., 2024). Recent shared tasks in machine translation and speech translation further show that quantization, pruning, and distillation can substantially reduce model size, while also revealing a persistent trade-off between compactness and translation quality (Gaido et al., 2025; Moslem, 2025; Moslem et al., 2025; Ponce et al., 2025).

This paper studies storage-oriented post-hoc compression in the constrained English-to-Chinese setting of the IWSLT 2026 Model Compression Track[1]. Starting from `Qwen2-Audio-7B-Instruct` as a fixed backbone, the paper evaluates a simple codec-based strategy that applies a lossy weight quantization step followed by lossless Zstandard coding only to selected transformer MLP projections, specifically `gate_proj`, `up_proj`, and `down_proj`, while leaving the remainder of the model unchanged. This design is motivated by the intuition that different components of a large speech translation model may not be equally compression-sensitive, and that targeting high-parameter feed-forward sublayers may preserve translation quality better than applying a uniform policy across all linear layers.

The proposed approach is compared against an 8-bit `bitsandbytes` baseline, a global codec baseline, and a more aggressive selected-layer pruning-plus-codec variant. The results show that targeted MLP compression is much more robust than global compression. In particular, selected-layer codec compression with `q3 + Zstd` achieves the highest COMET score among compressed systems, while `q2 + Zstd` provides the strongest overall quality-compression trade-off among the custom codec methods. Overall, the paper shows that simple

[1]Code available at https://github.com/alonsopg/iwslt-2026-model-compression.

Proceedings of the 23rd International Conference on Spoken Language Translation (IWSLT 2026), pages 40–46

layer-aware post-hoc compression can provide a practical alternative to more complex retraining-heavy compression pipelines in this shared-task setting.

2 Related Work

Model compression has become increasingly important in machine translation and speech translation because modern systems are often too large and computationally expensive for practical deployment. In multilingual machine translation, Mohammadshahi et al. (2022) show that post-training pruning and quantization can preserve average translation quality reasonably well, but may affect languages unevenly and amplify undesirable biases. This line of work highlights that compression should not be evaluated only by average performance, but also by how robustly quality is preserved under realistic variation.

In speech translation, the deployment challenge is intensified by long audio inputs and realistic test conditions. Salesky et al. (2023) introduce the ACL 60/60 evaluation sets, which emphasize long unsegmented audio, technical terminology, and other realistic difficulties that make efficient modeling especially important. More broadly, Gaido et al. (2024) survey the emerging landscape of speech foundation models and large language model based speech translation, emphasizing both the promise of these architectures and the practical limitations imposed by their computational cost. In parallel, Chen et al. (2024) show that strong end-to-end speech translation can be achieved with LLM-based systems through architectural design and training strategies, further underscoring the need for efficient deployment methods.

Compression-focused studies have become more direct in recent shared tasks. In the IWSLT 2025 Model Compression Track, Moslem (2025) investigate QLoRA, iterative layer pruning, and knowledge distillation for `Qwen2-Audio-7B-Instruct`, showing that substantial compression can retain most of the teacher model's translation quality. In the WMT 2025 Model Compression shared task, Gaido et al. (2025) report that quantization was the dominant strategy among submissions, while pruning and distillation were also used in several systems. Their overview includes methods based on GPTQ (Frantar et al., 2023), LeanQuant (Zhang and Shrivastava, 2025), iterative layer pruning (Moslem et al., 2025), and general knowledge distillation (Tan et al., 2023), as well as hybrid compression systems such as Vicomtech (Ponce et al., 2025). Taken together, these studies suggest that compression is now a central concern for deployable translation systems, but that existing methods still face a difficult trade-off between model compactness and translation quality, especially at stronger compression levels.

3 Task and dataset

This paper addresses the IWSLT 2026 Model Compression Track[2], specifically the *constrained* English-to-Chinese speech translation setting. The goal is to compress a speech translation system while preserving end-task translation quality. In this setting, the system maps spoken English directly to written Chinese, so compression must preserve both speech understanding and cross-lingual generation. Under the constrained condition, only the official ACL60/60 data may be used to support compression, including any data-dependent compression or post-compression adaptation step. The official track is built around `Qwen2-Audio` as the source model to be compressed.

Task. The target task is English-to-Chinese speech translation under the constrained track. No additional external data may be used to support model reduction. According to the shared-task definition, evaluation is based on translation quality, measured through automatic metrics, and model size on disk. Only the English-to-Chinese direction is considered in this study.

Data. All local experiments are conducted on the available English-to-Chinese evaluation split used for internal comparison. The split contains 416 utterances, each paired with an English source sentence and a Chinese reference translation. No additional audio segmentation is performed. Instead, the sentence-level segmented audio files distributed with the shared-task data are used; the local evaluation manifest points to files of the form `segmented_wavs/gold/sent_*.wav`. The official blind-test score is reported separately in Section 5.

Evaluation. Systems are evaluated in terms of translation quality and model size on disk. Translation quality is measured with COMET, a reference-based metric in which higher scores indicate better

[2] https://iwslt.org/2026/compression

quality. Specifically, `Unbabel/wmt22-comet-da` is used through `unbabel-comet` version 2.2.2, and the system-level COMET score is reported. COMET inputs consist of the English source text, the generated Chinese hypothesis, and the Chinese reference translation; scoring is run with batch size 16. Since the shared task explicitly targets compression, model size must be considered jointly with COMET.

Base model. The shared task defines `Qwen2-Audio-7B-Instruct` as the underlying model to be compressed[3]. According to its model card, `Qwen2-Audio` is an audio-language model that accepts audio inputs and supports both voice interaction and audio analysis through direct text generation. This makes it an appropriate backbone for the present task, where compression must preserve the ability to process spoken input while generating target-language text.

4 Compression Methods

All methods in this study are derived from the same backbone model, `Qwen2-Audio-7B-Instruct`, and are evaluated under the same inference conditions. The prompt, decoding configuration, evaluation split, and COMET-based assessment procedure are kept fixed across experiments so that observed differences can be attributed as much as possible to the compression method itself rather than to changes in the evaluation setup. In the custom codec-based variants, compression is applied only to the weight matrices of linear layers. Other components, including the tokenizer, non-linear activations, and the overall model architecture, remain unchanged. The objective is therefore not to redesign the model, but to examine how different compression policies over linear weights affect the trade-off between translation quality, inference efficiency, and storage reduction.

The custom codec pipeline consists of two separate stages: lossy weight quantization followed by lossless byte-level compression. First, the target linear layers are selected and their weight tensors are extracted. The weights are then quantized, which reduces numerical precision and introduces the only lossy step in the codec pipeline. Second, the quantized tensors are serialized and compressed with Zstandard. Zstandard is a general-purpose lossless compression algorithm that reduces storage size by encoding repeated and predictable byte patterns more efficiently. It does not change the numerical values of the tensor representation it receives. In this pipeline, Zstandard is applied after quantization, so it compresses the byte representation of the already quantized tensors rather than directly reducing model precision. At inference time, the byte stream is decompressed, the quantized tensor values are reconstructed exactly, and the reconstructed weights are used by a custom linear-layer wrapper. Consequently, the lossy quantization stage and the choice of compressed layers are the sources of quality degradation in the codec variants. Zstandard itself preserves the quantized representation exactly. Quantization can also make the tensor byte representation more compressible, because reduced numerical precision tends to increase redundancy and repeated byte patterns that Zstandard can encode efficiently.

Experimental Setup. All systems are evaluated under identical inference conditions in order to isolate the effect of compression. The shared backbone is `Qwen/Qwen2-Audio-7B-Instruct`, and all variants are tested on the official constrained English-to-Chinese evaluation split of 416 utterances. The prompt is fixed to *Translate this English speech into Chinese.*, generation uses `max_new_tokens=64`, and COMET is computed with the same `Unbabel/wmt22-comet-da` evaluation pipeline for all runs. Model compactness is measured by size on disk, and the reported comparisons therefore reflect the trade-off between translation quality, storage reduction, and inference efficiency under matched decoding conditions. Experiments were run on an Ubuntu 22.04 server with 8 CPU cores, 48.3 GB RAM, and one NVIDIA RTX A6000 GPU with 48 GB memory. No external data is used.

BitsAndBytes Int8 Quantization. The first compression baseline uses the `bitsandbytes` library to load the original model in 8-bit precision. This method does not alter the architecture of the model; instead, it replaces the standard floating-point weight representation with a lower-precision format supported by the library. This baseline is included because it is a widely adopted and easily reproducible reference point for practical large-model compression. It is directly integrated into the Hugging Face ecosystem and provides a useful

[3] https://huggingface.co/Qwen/Qwen2-Audio-7B-Instruct

benchmark against which more customized methods can be compared. In the present experiments, this approach preserves translation quality relatively well, but it does not improve runtime on the target hardware. Its main role in this study is therefore to serve as a stable standard baseline rather than as the final preferred deployment strategy.

Global Codec Compression. The first custom codec-based method applies the same compression rule to all linear layers of the model. This method is implemented with the `numcodecs` library (Alted et al., 2016), using its `Quantize`[4] filter and `Zstd`[5] codec. This codec quantization should be distinguished from deployment-oriented integer quantization methods such as int8 quantization: it is a lossy significant-digit quantization of floating-point weight values used for storage compression. For each linear weight matrix, the tensor is detached, moved to CPU, converted to a float32 NumPy array, and encoded with a pipeline consisting of `Quantize(digits=2, dtype="f4", astype="f2")` followed by `Zstd(level=3)`. The `Quantize` filter rounds floating-point weights according to the requested number of significant digits and stores the quantized representation using half precision. Zstandard then losslessly compresses the resulting quantized byte stream and reconstructs that quantized representation exactly during decompression. This method is included as the simplest custom codec baseline. Its main advantage is that it applies a single, uniform compression policy to all linear layers, which makes it straightforward to implement, reproduce, and analyze. At the same time, this design assumes that all linear layers tolerate the same compression strength, which is a strong assumption in a large multimodal model. The results indicate that this assumption is too coarse: applying this uniform lossy quantization to all linear layers substantially lowers translation quality, even though the accompanying Zstandard stage gives strong storage reduction. For that reason, this method serves primarily as a lower-bound custom baseline against which more selective compression strategies can be compared.

Selected-Layer Codec Compression. The second custom method relaxes the global assumption by compressing only a subset of linear layers. Rather than applying the same codec to all linear transformations, it targets the feed-forward sublayers in each transformer block, identified from the model hierarchy by names such as `mlp`, `feed_forward`, `up_proj`, `down_proj`, and `gate_proj`. These correspond to the MLP component of the transformer: `up_proj` expands the hidden representation, `down_proj` projects it back, and `gate_proj` implements the gating step in gated feed-forward variants. These layers are natural compression targets because they contain a large share of the model parameters, while perturbations in them may be less harmful to generation quality than perturbations in attention projections. All other linear layers remain unchanged. In `Qwen2-Audio-7B-Instruct`, this selection yields 96 compressed linear layers. Two codec settings are considered, `q2 + Zstd` and `q3 + Zstd`[6], where `q2` and `q3` denote shorthand for `Quantize(digits=2, dtype="f4", astype="f2")` and `Quantize(digits=3, dtype="f4", astype="f2")`, respectively, both followed by `Zstd(level=3)`. Thus, `q2` is the more aggressive setting, offering higher compression at the cost of greater information loss, whereas `q3` is milder and preserves more precision. This comparison tests whether stronger or weaker quantization yields the better quality-compression trade-off when applied only to selected layers. Unlike bitsandbytes, which replaces standard linear layers with specialized low-bit modules for model execution, this method applies generic lossy compression to stored weight tensors and reconstructs them at runtime. Overall, the results suggest that targeted layer selection is more effective than uniform global compression.

Selected-Layer Structured Pruning + Codec Compression. The final method combines layer selection with a more aggressive structural reduction strategy. As in the selected-layer codec variant, only a subset of feed-forward and MLP-style linear layers is targeted. Within those layers, structured pruning is first applied to remove part of the parameterization, after which codec compression is applied to the remaining weights. The motivation behind this design is to test whether combining two forms of reduction, structural removal and weight

[4] https://numcodecs.readthedocs.io/en/stable/filter/quantize.html

[5] https://numcodecs.readthedocs.io/en/stable/compression/zstd.html

[6] Compressed model artifacts for these variants are available at https://huggingface.co/alonsopg/qwen2audio-selected-layer-codec-q2 and https://huggingface.co/alonsopg/qwen2audio-selected-layer-codec-q3.

compression, can further improve compactness. In principle, this strategy offers a more aggressive alternative to codec-only compression. In practice, however, the reported results show that this combination leads to a much larger quality degradation than the codec-only selected-layer variants. It is therefore included as a comparison point illustrating that stronger reduction does not necessarily lead to a better quality-compression trade-off.

5 Results

Table 1 summarizes the compression results on the English-to-Chinese evaluation split. Here, *Sec/item* denotes average inference time per evaluated utterance in seconds, *Comp. Ratio* is the compression ratio relative to the FP16 baseline, and *Comp GB* reports the resulting model size on disk in gigabytes. The FP16 baseline obtains the highest COMET score, 0.7792, at 14.436 GB. Among the compressed systems, the selected-layer codec variant with q3 + Zstd achieves the best quality, reaching 0.7767 COMET, only 0.0025 below the baseline, at 10.312 GB. The q2 + Zstd variant is slightly lower in COMET, 0.7764, but reduces storage further to 8.422 GB, yielding the best quality-compression trade-off among the compressed systems. In contrast, applying lossy quantization to all linear layers lowers COMET to 0.7411, while the subsequent Zstandard stage helps reduce the stored model size to 3.628 GB. The structured pruning plus codec compression variant performs worst, at 0.5871 COMET.

Overall, the results show that compression quality depends strongly on where compression is applied. Restricting compression to selected MLP projections preserves translation quality much better than a uniform global policy, suggesting that these feed-forward layers provide a more favorable compression target. The comparison between q2 and q3 highlights the expected trade-off: q3 gives the best absolute quality among compressed models, whereas q2 provides the strongest balance between quality and size. The bitsandbytes int8 baseline remains competitive, but in this setup it is both slower and less storage-efficient than the selected-layer codec variants.

For the final submission file, a small amount of conservative post-processing was applied to remove explicit meta-translation prefixes and a few obvious non-Chinese leakage patterns. This cleanup did not affect the underlying model or compression method, and Table 1 should therefore be read as a comparison of core model variants rather than post-hoc output normalization.

Submission. The submitted system, `APG_IWSLT26_ModelCompression_en-zh_constrained_primary`, is based on `Qwen/Qwen2-Audio-7B-Instruct` and applies selected-layer codec compression to transformer MLP projections, specifically `gate_proj`, `up_proj`, and `down_proj`. The primary run uses q3 + Zstd over 96 selected layers. Although q2 + Zstd provides the strongest overall quality-compression trade-off, q3 + Zstd was selected as the primary submission because it achieves the highest COMET score among the compressed systems. The system was submitted through the official SPEECHM evaluation platform[7].

Official blind-test score. After submission, the IWSLT 2026 organizers reported an official ACL60/60 blind-test COMET score of 0.339 for the primary constrained English-to-Chinese run, `APG_IWSLT26_ModelCompression_en-zh_constrained_primary`. This score is included for completeness as the official shared-task result. Because the blind-test evaluation and the local ablation experiments were conducted under different evaluation conditions, they are reported separately: the official score reflects the submitted primary run on the blind test set, while Table 1 reports matched local comparisons among compression variants. The shared task and official campaign are described by Adelani et al. (2026).

6 Conclusions

This paper examined storage-oriented post-hoc compression for constrained English-to-Chinese speech translation in the IWSLT 2026 Model Compression Track, using `Qwen2-Audio-7B-Instruct` as the fixed backbone. The main focus was selected-layer codec compression, where lossy quantization followed by lossless Zstandard coding is applied only to transformer MLP projections, specifically `gate_proj`, `up_proj`, and `down_proj`, while the rest of the model remains unchanged. This design was compared against an FP16 baseline, an 8-bit `bitsandbytes` baseline, a global codec baseline, and a more aggressive selected-layer pruning-plus-codec variant.

[7] https://speechm.cloud.cyfronet.pl/

#	Method	COMET	Δ	Sec/item	Comp. Ratio	Comp GB	Setting
1	FP16 baseline	**0.779163**	+0.000000	**0.664055**	1.000	14.436	fp16
2	BitsAndBytes Int8	0.775192	-0.003971	2.538633	1.733	9.026	int8
3	Selected-layer codec	0.776689	-0.002473	0.751678	1.400	10.312	q3 + zstd-3
4	**Selected-layer codec**	**0.776365**	**-0.002798**	**0.745540**	**1.714**	**8.422**	**q2 + zstd**
5	Global codec	0.741091	-0.038071	0.836255	3.979	3.628	q2 + zstd
6	Sel.-layer pruning + codec	0.587115	-0.192048	0.856106	1.800	8.020	prune + q2

Table 1: Compression results on English-to-Chinese speech translation with Qwen2-Audio. Δ denotes the absolute COMET change relative to the FP16 baseline. Selected-layer codec compression with q2 + Zstd provides the best quality-compression tradeoff among the compressed variants.

The local controlled results show that where compression is applied matters substantially. Applying lossy quantization to all linear layers causes a clear drop in translation quality, even though the subsequent Zstandard stage contributes strong storage reduction. In contrast, restricting compression to selected MLP projections preserves performance much more effectively, indicating that these layers are a more favorable target for compact post-hoc compression. Among the compressed systems, the `q3 + Zstd` selected-layer variant achieves the highest COMET score, remaining very close to the FP16 baseline, while the `q2 + Zstd` variant provides the strongest overall balance between quality and storage reduction among the custom codec methods. The pruning-plus-codec comparison further shows that more aggressive reduction does not necessarily lead to a better quality-compression trade-off. The official ACL60/60 blind-test COMET score reported for the submitted primary constrained run was 0.339, and this result is included separately as the official shared-task score.

Overall, these findings suggest that simple layer-aware codec compression can serve as an effective and practical alternative to more complex retraining-based compression pipelines for speech translation. In particular, the study shows that competitive compression can be achieved without distillation, architectural redesign, or extensive post-compression adaptation. This makes the approach appealing in constrained settings where simplicity, reproducibility, and low implementation overhead are important.

A natural direction for future work is to extend layer-aware compression beyond a fixed hand-selected subset and investigate finer-grained policies that adapt compression strength across components of the model. It would also be valuable to evaluate whether the same pattern holds across other language pairs, datasets, and speech translation backbones, and whether targeted compression can be combined with modest adaptation techniques to further improve the quality-compression trade-off.

Declaration of Generative Writing Assistance

Grammarly[8] was used during the preparation of this paper for writing assistance, language improvement, and text condensation. The author reviewed and edited the final text and remains responsible for the paper's content.

References

David Ifeoluwa Adelani, Antonios Anastasopoulos, Victor Agostinelli, 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, Danni Liu, Nam Luu, Min Ma, Dominik Machaček, Marie Maltais, Evgeny Matusov, Chandresh Kumar Maurya, John P. McCrae, Yasmin Moslem, Satoshi Nakamura, Matteo Negri, Jan Niehues, Atul Kr. Ojha, Siqi Ouyang, Sara Papi, Peter Polák, Fabian Retkowski, Beatrice Savoldi, Claytone Sikasote, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Alex Waibel, Patrick Wilken, Vilém Zouhar, and Maike Züfle. 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.

Francesc Alted, Prakhar Goel, Jerome Kelleher, John Kirkham, Alistair Miles, Jeff Reback, Trevor Manz,

[8] https://www.grammarly.com/

Grzegorz Bokota, Josh Moore, Martin Durant, and Paul Branson. 2016. Numcodecs. https://numcodecs.readthedocs.io/en/stable/. Python package for buffer compression and transformation codecs.

Xi Chen, Songyang Zhang, Qibing Bai, Kai Chen, and Satoshi Nakamura. 2024. LLaST: Improved end-to-end speech translation system leveraged by large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6976–6987, Bangkok, Thailand. Association for Computational Linguistics.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*.

Marco Gaido, Roman Grundkiewicz, Thamme Gowda, and Matteo Negri. 2025. Findings of the WMT 2025 shared task on model compression: Early insights on compressing LLMs for machine translation. In *Proceedings of the Tenth Conference on Machine Translation*, pages 484–494, Suzhou, China. Association for Computational Linguistics.

Marco Gaido, Sara Papi, Matteo Negri, and Luisa Bentivogli. 2024. Speech translation with speech foundation models and large language models: What is there and what is missing? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14760–14778, Bangkok, Thailand. Association for Computational Linguistics.

Alireza Mohammadshahi, Vassilina Nikoulina, Alexandre Berard, Caroline Brun, James Henderson, and Laurent Besacier. 2022. What do compressed multilingual machine translation models forget? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4308–4329, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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, Vienna, Austria (in-person and online). Association for Computational Linguistics.

Yasmin Moslem, Muhammad Hazim Al Farouq, and John Kelleher. 2025. Iterative layer pruning for efficient translation inference. In *Proceedings of the Tenth Conference on Machine Translation*, pages 1022–1027, Suzhou, China. Association for Computational Linguistics.

David Ponce, Harritxu Gete, and Thierry Etchegoyhen. 2025. Vicomtech@WMT 2025: Evolutionary model compression for machine translation. In *Proceedings of the Tenth Conference on Machine Translation*, pages 1011–1021, Suzhou, China. 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, Toronto, Canada (in-person and online). Association for Computational Linguistics.

Shicheng Tan, Weng Lam Tam, Yuanchun Wang, Wenwen Gong, Shu Zhao, Peng Zhang, and Jie Tang. 2023. GKD: A general knowledge distillation framework for large-scale pre-trained language model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 134–148, Toronto, Canada. Association for Computational Linguistics.

Tianyi Zhang and Anshumali Shrivastava. 2025. LeanQuant: Accurate and scalable large language model quantization with loss-error-aware grid. In *The Thirteenth International Conference on Learning Representations*.