

Pare4Bit: Quantization, Depth Pruning, and Model Selection for the WMT26 Model Compression Task

Nail Fakhrutdinov

Independent

fano2458@gmail.com

Abstract

We describe Team Pare4Bit’s submissions to the WMT26 Model Compression shared task on the three evaluation pairs Czech→German, English→Simplified Chinese, and English→Egyptian Arabic. In the *constrained* track we compress google/gemma-3-12b-it with a data-light pipeline—vision-tower removal followed by GPTQ W4A16 quantization—reaching 43.2 mean chrF at 7.6 GB, $3.1\times$ smaller than the bf16 text-only reference for a 0.3 chrF drop. We further explore depth pruning with LoRA knowledge-distillation healing, which recovers a collapsed pruned model to within 1.3 chrF of the INT4 primary and adds an engine-agnostic speedup. In the *unconstrained* track we treat model selection as a first-class compression decision: the newer Gemma-4-12B, deployed via quantization-aware-trained INT4, outperforms the constrained Gemma-3 baseline by +2.5 mean chrF (+4.7 on Egyptian Arabic, where it produces the dialect rather than Modern Standard Arabic). We report several negative results—activation quantization instability on Gemma-3, re-segmentation under naive vocabulary pruning, and a chrF-versus-COMET disagreement on a sentence-level MT specialist applied to paragraphs—as practical guidance on what does and does not compress this model family.

1 Introduction

The WMT26 Model Compression task asks participants to shrink a large translation model while preserving quality, ranking systems on a Pareto frontier of quality, on-disk/VRAM size, and inference speed (Gowda et al., 2026). We participate in both tracks on all three pairs (ces→deu, eng→zho_Hans, eng→ara_EG).

Our guiding hypothesis, borne out by our experiments, is that Gemma-3-12B is substantially over-parameterized for these translation directions,

so aggressive weight quantization is nearly free while structural changes (depth pruning) trade quality gradually. Our contributions are:

- A clean, reproducible constrained pipeline (vision-tower removal + GPTQ W4A16) that loses only 0.3 chrF at $3.1\times$ compression relative to the bf16 text-only reference (§4).
- A depth-pruning + LoRA knowledge-distillation healing study that maps the quality/speed knee and shows healing recovers an otherwise-collapsed model (§5).
- An unconstrained entry showing that *choosing the newest sub-20B model* (Gemma-4-12B, quantization-aware-trained (QAT) INT4) beats the constrained baseline outright, with a dialect-fidelity finding on Egyptian Arabic (§6).
- A set of negative results delimiting what fails on this model family (§8).

2 Related Work

Post-training quantization (PTQ) shrinks weights without retraining: GPTQ (Frantar et al., 2023) uses second-order (Hessian) information to quantize weights to 4 bits with per-group scales, AWQ (Lin et al., 2024) protects salient channels identified from activation statistics, and SmoothQuant (Xiao et al., 2023) migrates activation outliers into the weights so that activations can also be quantized. Pruning removes parameters instead of narrowing them: SparseGPT (Frantar and Alistarh, 2023) performs one-shot unstructured and 2:4-structured sparsification, while *depth* pruning drops whole decoder layers ranked by redundancy—ShortGPT’s Block Influence (Men et al., 2025) is the ranking we adopt. Aggressive removal typically needs recovery training; we use low-rank adaptation (Hu et al., 2022) for a lightweight, distillation-based heal. For evaluation we rank with chrF (Popović,

2015) computed by sacreBLEU (Post, 2018) and cross-check with COMET (Rei et al., 2020). Our contribution is not a new algorithm but a careful, reproducible application of these tools to one 12B multilingual model, together with an accounting of what fails on it.

3 Experimental Setup

Models and serving. The constrained base is gemma-3-12b-it (12B parameters, 48 decoder layers, hidden size 3840, a 262k-entry tied input/output embedding of ~ 2 GB in bf16; ~ 24 GB total). All quantized systems are served with vLLM 0.24 through the compressed-tensors format and Marlin kernels; the bf16 reference runs unquantized in the same engine. We serve text-only: for Gemma-3 we rebuild a Gemma3ForCausalLM from the multimodal checkpoint’s text_config, discarding the 417M-parameter vision tower and its 4.4M projector. Because the chat template already emits `<bos>`, we prompt through the chat API to avoid a duplicated beginning-of-sequence token.

Data. Calibration and healing use only WMT news_commentary v18.1 for the three pairs, taken from the constrained-track recipe. GPTQ is calibrated on 256 chat-formatted samples truncated to 1024 tokens; healing uses 18k teacher-generated targets (6k per pair, source lines filtered to 20–600 characters; §5). No development or test data is used for calibration or training.

Hardware and batching. Both constrained experiments and unconstrained evaluation run on a single A100. vLLM continuous batching is essential to a fair speed comparison: throughput rises $42\times$ from batch 1 to batch 64 and saturates near 64 (batch 512 adds only $\sim 7\%$ while holding $8\times$ as many sequences in the KV cache), so we report speed at batch 64 and declare a recommended maximum batch of 64–128.

Evaluation. We report chrF (Popović, 2015) via sacreBLEU (Post, 2018) on the WMT25 development sets with greedy decoding and rank by it, the task metric; COMET (Rei et al., 2020) (wmt22-comet-da, reference implementation) is a secondary metric. Speed is decode throughput at batch 64 with a fixed output length (Table 2); end-to-end wall-clock on our shared host varied by up to 13% between sessions for identical models, so we do not rank on it.

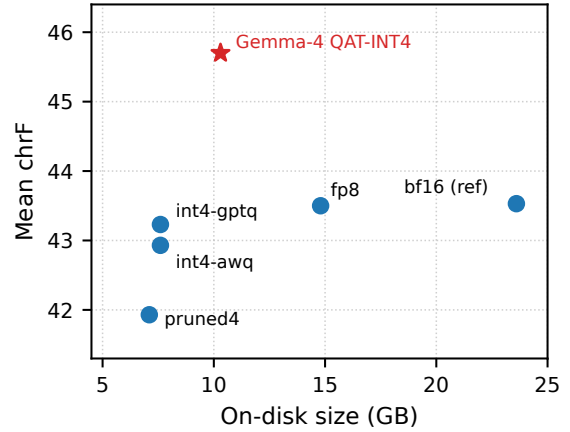


Figure 1: Size against quality. int4-gptq keeps almost all quality at a third of the size; Gemma-4 (star) has the highest quality but a larger footprint. Heavily pruned points are shown in Figure 2.

4 Constrained Track

Primary: GPTQ W4A16. Our primary system, int4-gptq, applies GPTQ (Frantar et al., 2023) to every Linear layer with 4-bit weights, group size 128, and 16-bit activations; the tied lm_head/embedding is excluded and kept in bf16. Table 1 shows it preserves quality (43.23 vs. 43.53 mean chrF for the bf16 text-only reference) at 7.6 GB—a $3.1\times$ shrink over the text-only reference (and $3.2\times$ over the full multimodal checkpoint) for 0.3 chrF. The cost is uneven across directions: it is significant only on eng-zho. Figure 1 places all systems on the size–quality plane.

Contrastives. FP8 dynamic quantization is data-free and near-lossless but larger (14.8 GB); on A100, which has no FP8 tensor cores, its throughput is 16% above bf16 but 4% below INT4 (Table 2). AWQ (Lin et al., 2024) trails GPTQ slightly (42.93, significant only on ces-deu): llmcompressor skipped 48 of its inferred scaling mappings on Gemma-3 for incompatible shapes, one per decoder layer by count, so part of each layer is quantized without activation-aware scaling; we did not identify which mapping.

5 Depth Pruning with Distillation Healing

Because quantization barely dents quality, we ask whether *structural* compression can add speed. We score each decoder layer by Block Influence (Men et al., 2025), $1 - \cos(\text{input}, \text{output})$ averaged over calibration data, and drop the N lowest-influence layers. Block Influence selects layers {15, 16, 19, 20}

System	Size (GB)	chrF				COMET
		eng-zho	ces-deu	eng-ara	mean	mean
bf16 text-only (ref.)	23.6	28.4*	63.6	38.6	43.53	0.797
int4-gptq (primary)	7.6	28.1	63.4	38.2	43.23	0.793
fp8	14.8	28.3	63.4	38.8*	43.50	0.798
int4-awq	7.6	28.1	62.9*	37.8	42.93	0.795
pruned4-healed-int4	7.1	27.0*	60.8*	38.0	41.93	0.788
pruned6-healed-int4	6.9	23.7*	53.7*	34.1*	37.17	0.748
pruned12-healed-int4	6.2	14.9*	39.5*	27.8*	27.40	0.640
Gemma-4-12B QAT-INT4 (unconstr.)	10.3	29.0*	65.2*	42.9*	45.70	0.811

Table 1: chrF and COMET (wmt22-comet-da, mean over the three pairs) on the WMT25 development sets (greedy). *: chrF significantly different from int4-gptq (paired bootstrap, 1000 resamples, $p < 0.05$; 95% CIs ± 0.5 to 1.3 chrF). Depth-pruned rows are the submitted healed INT4 models.

for $N=4$, {15,16,18,19,20,22} for $N=6$ and {14-20, 22-26} for $N=12$: clustered in the middle third of the network, not strictly contiguous. Dropping layers shortens the sequential decode path, so unlike weight quantization it lowers latency on *any* engine.

Removing layers without repair is catastrophic: dropping 12 of 48 layers collapses quality to ~ 6 mean chrF (near-random). We heal with a LoRA adapter (Hu et al., 2022)—rank 16, $\alpha=32$, dropout 0.05, applied to all seven attention and MLP projection matrices (q, k, v, o, gate, up, down)—trained on the 18k teacher targets with a completion-only loss (prompt tokens masked). We optimize with AdamW at 10^{-4} under a cosine schedule (3% warmup) for two epochs, effective batch size 16, sequences to 1024 tokens, then merge the adapter and re-quantize with GPTQ. Teacher targets are greedy decodes from the uncompressed text-only Gemma-3-12B using the same prompt template as inference, so the student learns to imitate the uncompressed model directly.

Healing recovers the 12-layer model from collapse to 27.4 mean chrF and the 4-layer model to 41.9 (Table 1), 1.3 below int4-gptq and 1.6 below the bf16 reference. Applying INT4 to the healed weights is not free: it costs a further 0.3 chrF at 4 layers and 1.0 at 6. Figure 2 traces the trade-off: quality degrades roughly 2 chrF per dropped layer past a ~ 4 -layer knee, and ces-deu (the highest-scoring pair) is the most sensitive. The 4-layer variant (pruned4-healed-int4, 44 layers) is our fast/small contrastive: with output length held fixed, its decode throughput at batch 64 is 9% higher than int4-gptq, and that of the 6- and 12-layer variants 14% and 32% higher (Table 2); step time falls in proportion to the layers removed. End-

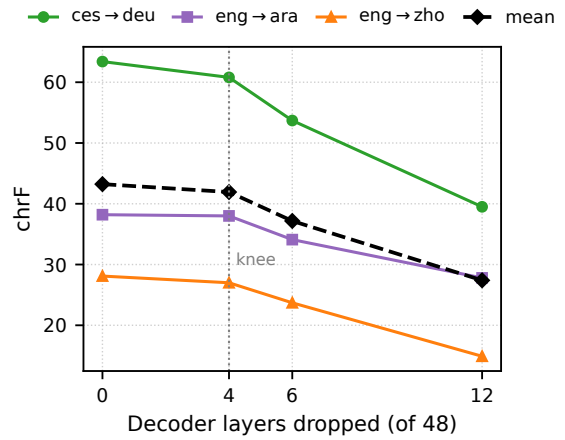


Figure 2: Depth pruning after healing: chrF against dropped decoder layers. Quality is stable to about 4 layers (dotted line), then falls roughly 2 chrF per layer; ces-deu is the most sensitive pair.

to-end time falls less than throughput rises, because the 6- and 12-layer models also over-generate: they emit 18% and 26% more tokens than the primary (Tokens column in Table 2), often repeating sentences, and 13% of the 12-layer outputs run into the length cap; chrF barely charges for the surplus text itself (truncating it to the reference length moves chrF by under 1 point); the loss it registers is the content omitted while looping. Peak VRAM is uninformative on this stack: vLLM reserves a fixed 92% of the card for every system, so Table 2 reports the weight memory it loads instead.

6 Unconstrained Track: Model Selection as Compression

The unconstrained track permits any model under 20B parameters. We treat model *choice* as the primary lever and survey the newest sub-20B releases.

System	Layers	Disk (GB)	Weights (GB)	Tokens (k)	Decode (tok/s)
bf16 text-only	48	23.6	24.3	80.7	1421
int4-gptq	48	7.6	8.4	80.7	1730
fp8	48	14.8	13.6	80.8	1653
int4-awq	48	7.6	8.5	80.6	1712
pruned4-healed-int4	44	7.1	7.9	77.7	1884
pruned6-healed-int4	42	6.9	7.7	94.8	1974
pruned12-healed-int4	36	6.2	7.0	101.7	2284

Table 2: Size and speed on one A100-80GB. Disk: checkpoint size; Weights: model memory reported by vLLM; Tokens: output tokens on the ces-deu blind set (457 paragraphs, batch 64, median of three runs); Decode: throughput at batch 64 with 256 forced output tokens per prompt (median of two runs, spread below 1%).

System	chrF			
	eng-zho	ces-deu	eng-ara	mean
Gemma-3 bf16	28.4	63.6	38.6	43.5
Gemma-3 int4-gptq	28.1	63.4	38.2	43.2
Gemma-4 bf16	29.6	65.6	43.2	46.1
Gemma-4 QAT-INT4	29.0	65.2	42.9	45.7

Table 3: Unconstrained track: Gemma-4-12B (bf16 and the submitted QAT-INT4) against Gemma-3-12B (bf16 text-only and the constrained INT4 primary). Sizes in Table 1; Gemma-4 bf16 is 24 GB.

Gemma-4-12B (released after WMT25) markedly outperforms Gemma-3-12B on all three pairs (Table 3). We deploy Google’s QAT INT4 checkpoint (W4A16, compressed-tensors), which loads directly in vLLM. Against our own bf16 evaluation of Gemma-4-12B it costs 0.4 mean chrF (significant on all three pairs), about what GPTQ costs Gemma-3 (0.3), so the gain over the constrained primary comes from the model, not from QAT.

The gain concentrates on English→Egyptian Arabic (+4.7 chrF): inspection shows Gemma-4 produces the Egyptian dialect where weaker systems fall back to Modern Standard Arabic, which is penalized against the dialectal reference. COMET agrees: Gemma-4’s Arabic gain over the primary is also its largest under COMET (+0.025 against +0.014 on the other pairs), so it is not only surface matching of the dialectal reference. At ~10 GB on disk it is larger than our constrained INT4 primary but wins on quality; we verified it runs text-only from a clean clone.

7 Analysis

Gemma-3-12B is over-parameterized for these directions. Two independent observations converge on the same conclusion. Weight-only 4-bit quantization costs only 0.3 chrF, and four decoder layers can be removed for a further 1.3 chrF once

healed (1.6 below the bf16 reference). The model carries far more capacity than these three translation directions require, which is why compression is nearly free until structural pruning eventually crosses the knee.

The right lever depends on the bottleneck. If the constraint is storage or VRAM, weight-only INT4 is the clear pick (~3× smaller, near-lossless). If it is single-stream latency, quantization does *not* help on A100: int4-gptq is 33% slower than bf16 at batch 1 even though its throughput is 22% higher at batch 64 (§8, Table 2), so depth pruning is the only speedup that holds in both regimes (9% at four layers, 32% at twelve). A compression entry is therefore best described by *which* axis of the Pareto surface it targets.

Model choice can dominate compression. Given freedom to pick any sub-20B model, adopting the newer Gemma-4-12B recovers more quality (+2.5 mean chrF) than any of our compression steps costs, at a modest size increase (Figure 1). The Egyptian-Arabic results also show that the *reference dialect* drives chrF but not COMET: the Hunyuan-MT specialist of §8 outputs Modern Standard Arabic, which costs 8 chrF against the primary on short paragraphs at unchanged COMET, while Gemma-4’s dialectal output gains 4.7 chrF that COMET confirms as a genuine improvement.

8 Negative Results

We found several intuitive ideas do not help on this stack, and report them as guidance.

- **Activation quantization is unstable on Gemma-3.** W8A8-INT8 (with SmoothQuant (Xiao et al., 2023)) collapses to ~7 mean chrF; W4A8 is blocked by a packing-format mismatch between our quantizer and serving kernels.

- **Naive vocabulary pruning re-segments text.** Dropping unused tokens, keeping their merge ancestors, removing merges that touch dropped tokens, and renumbering still re-segmented $\sim 7\%$ of dev lines under greedy BPE, mostly Arabic. Our merge closure kept one parent pair per token, so tokens reachable by several merges could lose the path greedy BPE takes; a stricter rule (never remove a token that is an input of a surviving merge) should fix this. We scope the result to our implementation; the round-trip gate that caught it is the reusable lesson.
- **Sentence-level MT specialists lose on chrF, not on COMET.** Hunyuan-MT-1.5-7B, a WMT25 metrics-winning specialist, scores 35.7 mean chrF against 43.2 for our primary, yet COMET rates it above every Gemma system on eng-zho (0.841 vs 0.814) and ces-deu (0.821 vs 0.796): its renderings are longer and freer, which chrF penalizes on paragraphs and COMET rewards. On eng-ara it loses under both metrics (21.3 chrF, 0.691 COMET): it outputs Modern Standard Arabic and, on long paragraphs, about half the reference length.
- **Weight-only INT4 speeds batched decode, not single-stream decode.** On A100, int4-gptq is 33% slower than bf16 at batch 1 (9316 vs 6990 s), where Marlin dequantization dominates; at batch 64, with output length held fixed, its throughput is 22% higher than bf16 (Table 2). Its wins are footprint, KV-cache headroom and batched throughput, not single-stream latency.

9 Conclusion

Aggressive weight quantization compresses Gemma-3-12B for MT almost for free; structural pruning trades quality gradually and needs distillation healing; and, given freedom of model choice, simply adopting the newest strong base is the most effective “compression” of all. Code and models are available.¹

References

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 (ICML)*, volume

202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. [GPTQ: Accurate post-training quantization for generative pre-trained transformers](#). In *International Conference on Learning Representations (ICLR)*.

Thamme Gowda, Marco Gaido, Roman Grundkiewicz, and Matteo Negri. 2026. Findings of the WMT 2026 shared task on model compression: No free lunch at extreme compression. In *Proceedings of the Eleventh Conference on Machine Translation (WMT 2026)*, Budapest, Hungary.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations (ICLR)*.

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)*.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2025. [ShortGPT: Layers in large language models are more redundant than you expect](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20192–20204, Vienna, Austria. Association for Computational Linguistics.

Maja Popović. 2015. [chrF: character n-gram F-score for automatic MT evaluation](#). In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. 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.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. [SmoothQuant: Accurate and efficient post-training quantization for large language models](#). In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 38087–38099. PMLR.

¹<https://github.com/fano2458/wmt26-pare4bit>