

Vicomtech@WMT 2026: Mask-based Pruning for Model Compression

David Ponce^{1,2} and Thierry Etchegoyhen¹

¹ Fundación Vicomtech, Basque Research and Technology Alliance (BRTA)

² University of the Basque Country EHU

{adponce, tetchegoyhen}@vicomtech.org

Abstract

We describe Vicomtech’s participation in the constrained track of the WMT 2026 Shared Task on Model Compression. We addressed all three translation directions of the task, namely Czech to German, English to Simplified Chinese, and English to Egyptian Arabic, using Gemma 3 12B IT as our base model. Our approach performs task-aware structured post-training pruning, which learns differentiable masks to prune model components using machine translation calibration data and an objective combining next-token cross-entropy, knowledge distillation, and intermediate representation matching. We systematically evaluated pruning at two target compression ratios (0.25 and 0.50) and explored different structured pruning configurations, combined with supervised fine-tuning to recover performance after pruning. Additionally, we explored language-specific vocabulary pruning and 4-bit AWQ quantization to achieve further reductions in model size and memory requirements. Our experimental results demonstrate that the combination of structured pruning, targeted post-training, vocabulary reduction, and quantization can achieve substantial model compression while maintaining competitive translation quality across all language directions.

1 Introduction

Large Language Models (LLMs) have become a central paradigm in natural language processing, demonstrating strong capabilities across a wide range of language understanding and generation tasks (Radford et al., 2019; Brown et al., 2020; Chang et al., 2024). Their impact has increasingly extended to machine translation (MT), where general-purpose LLMs can achieve results comparable to, and in some settings surpassing,

those of specialized neural machine translation systems (Xu et al., 2024; Zhu et al., 2024; Kocmi et al., 2023, 2024). This progress has also encouraged the development of multilingual and general-purpose model families with strong cross-lingual capabilities, such as Aya Expanse (Dang et al., 2024), EuroLLM (Martins et al., 2025), and Tower (Alves et al., 2024).

However, deploying these models remains computationally demanding. Their large parameter counts require substantial memory and processing resources during both training and inference, while the associated energy consumption raises broader sustainability concerns (Strubell et al., 2019). These demands limit their practical deployment, particularly in settings where reduced memory usage and efficient inference are essential.

In this work, we describe Vicomtech’s participation in the constrained track of the WMT 2026 Model Compression shared task (Gowda et al., 2026). The task focuses on making LLM-based machine translation systems suitable for deployment in resource-constrained environments by balancing translation quality, memory requirements, and inference efficiency. Participants in the constrained track were tasked with compressing Gemma 3 12B IT (Gemma Team, 2025) while maintaining competitive translation performance across three translation directions: Czech to German, English to Simplified Chinese, and English to Egyptian Arabic.

To address this challenge, we developed a task-aware structured post-training pruning approach based on differentiable masks learned from machine translation calibration data. We systematically evaluated different structured pruning configurations at target compression ratios of 0.25 and 0.50 and applied supervised fine-tuning to recover performance after pruning. Additionally, we explored language-specific vocabulary pruning and quantization methods to achieve further reductions

in model size and memory requirements.

2 Related Work

Large language models have become increasingly competitive in machine translation, often matching or exceeding dedicated neural MT systems (Kocmi et al., 2023, 2024). This progress has encouraged the development of multilingual models with strong translation and cross-lingual capabilities, including Aya Expanse (Dang et al., 2024), EuroLLM (Martins et al., 2025), and Tower (Alves et al., 2024). However, their substantial memory and computational requirements continue to limit practical deployment. Efficient model compression is therefore particularly important for machine translation, where systems must balance translation quality with model size, memory consumption, and inference efficiency.

Traditional compression techniques for large language models include quantization, knowledge distillation, and pruning. Quantization reduces the numerical precision of model parameters to lower storage and memory requirements, during training or as a post-training optimization process (Whitaker and Raj, 2001; Xiao et al., 2023; Dettmers et al., 2023). Of direct relevance to this work, AWQ (Lin et al., 2024) uses activation statistics to identify salient weight channels and applies selective scaling to reduce low-bit quantization errors.

Among pruning approaches, structured methods that remove complete layers, channels, or attention components are especially attractive because they produce smaller dense models that can be executed efficiently on standard hardware. SliceGPT (Ashkboos et al., 2024) uses PCA-based rotations to identify low-importance hidden dimensions, which are then removed by deleting the corresponding rows and columns of the weight matrices, while LLM-Pruner (Ma et al., 2023) uses gradient information to identify prunable structures. LaCo (Yang et al., 2024) instead compresses models by merging consecutive layers according to cosine similarity differences. FLAP (An et al., 2024) introduces a retraining-free framework based on fluctuation-aware importance estimation, adaptive global structure selection, and bias compensation to mitigate the performance degradation caused by pruning. GRASPrune (Wang et al., 2026) jointly removes feed-forward channels and grouped-query attention KV head groups under a single global parameter budget.

It learns continuous gate scores on a small calibration set and projects them at each step to a hard mask satisfying the budget, while keeping the backbone parameters frozen.

Search-based methods have also been explored for model compression. EvoPress (Sieberling et al., 2024) formulates compression as a general optimization problem and uses evolutionary search to identify dynamic, non-uniform compression configurations. GeLaCo (Ponce et al., 2025a) applies evolutionary search to identify effective layer-collapse configurations; this approach was notably employed in the WMT 2025 translation model compression shared task (Ponce et al., 2025b).

Our approach is related to GRASPrune, being based on learning differentiable masks over structured pruning units, keeping the backbone parameters frozen and optimizing continuous mask logits, which are converted into hard masks during the forward pass while remaining trainable through a straight-through estimator. In this work, the masks are learned on an MT-specific calibration set, using the original unmasked base model as a teacher. We further enforce a uniform structure across layers, retaining the same FFN width and the same number of GQA groups in every layer. FFN widths are additionally constrained to multiples of 128 to ensure compatibility with efficient inference engines such as vLLM and hardware-aligned tensor dimensions.

3 Methodology

Our approach combines task-aware structured pruning with recovery and complementary compression techniques. As illustrated in Figure 1, we attach learnable masks to selected feed-forward and attention components while keeping the parameters of the base model frozen. Using the original model as teacher, we optimize the pruning structure using MT calibration examples. Finally, the best performing learned mask is fixed and the corresponding components are pruned. We also explored three additional optimization strategies: supervised fine-tuning as a post-pruning recovery strategy, along with quantization and vocabulary pruning for further compression.

3.1 Structured Mask Pruning

Pruning units. We consider two types of structured pruning units: feed-forward network (FFN)

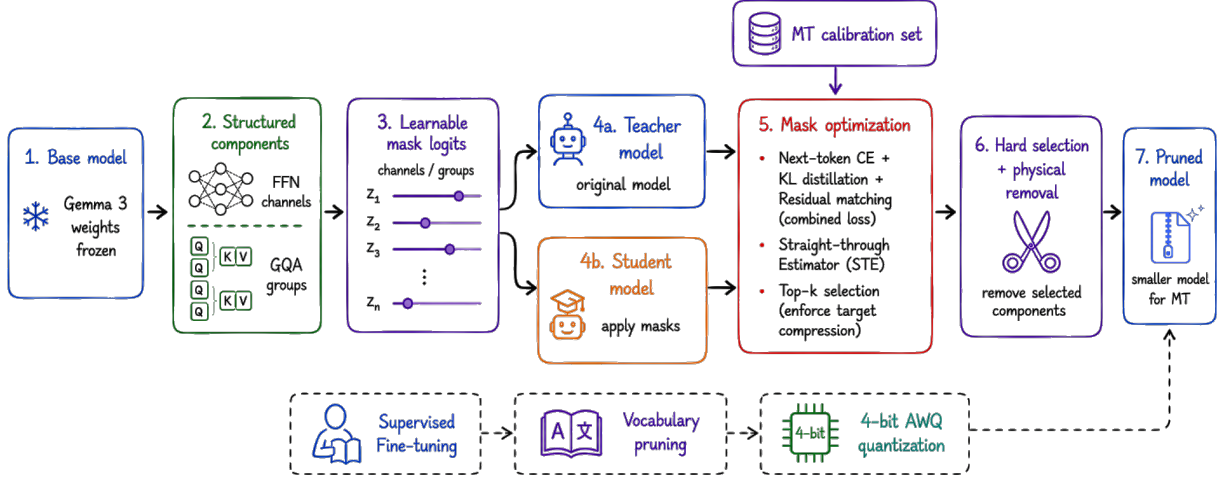


Figure 1: Overview of the proposed model compression pipeline. Learnable masks are attached to FFN channels and GQA groups of the Gemma 3 12B IT model and optimized using MT data as calibration. The masks are trained using a combination of next-token cross-entropy and knowledge distillation. The resulting model can then undergo SFT for performance recovery, 4-bit AWQ quantization and vocabulary pruning for further optimization.

channels and grouped-query attention (GQA) groups. Before introducing the masking mechanism, we describe the computations associated with these units and establish the required notation.

For the gated FFN at layer ℓ , the intermediate representation is described in Equation (1):

$$\mathbf{u}_\ell(\mathbf{h}) = \phi(W_\ell^{\text{gate}}\mathbf{h}) \odot W_\ell^{\text{up}}\mathbf{h} \quad (1)$$

where ϕ denotes the model’s activation function and $\odot$ denotes element-wise multiplication. Each intermediate dimension defines one FFN pruning unit, jointly comprising the corresponding rows of W_ℓ^{gate} and W_ℓ^{up} and the matching column of W_ℓ^{down} . The FFN output is indicated in Equation (2):

$$\text{FFN}_\ell(\mathbf{h}) = W_\ell^{\text{down}}\mathbf{u}_\ell(\mathbf{h}) \quad (2)$$

For attention, GQA organizes the query, key, and value projections so that multiple query heads share the same key and value heads. Given the hidden representation $\mathbf{h}$, the query, key, and value projections produce $\mathbf{Q}_\ell$, $\mathbf{K}_\ell$, and $\mathbf{V}_\ell$, respectively. The query projection is divided into H_q heads, whereas the key and value projections are divided into H_{kv} heads.

Let $g(r)$ denote the key-value group assigned to query head r . The output of that query head is established as in Equation (3):

$$\mathbf{a}_{\ell,r}(\mathbf{h}) = \text{Softmax}\left(\frac{\mathbf{Q}_{\ell,r}\mathbf{K}_{\ell,g(r)}^\top}{\sqrt{d_h}}\right)\mathbf{V}_{\ell,g(r)} \quad (3)$$

where d_h is the head dimension. For each key-value pair, the outputs of its associated query heads are concatenated to form one GQA-group output. We denote the ordered collection of these group outputs by $\mathbf{A}_\ell(\mathbf{h})$. One GQA pruning unit consists of one key-value head together with its associated query heads. The attention transformation is described in Equation (4):

$$\text{Attn}_\ell(\mathbf{h}) = \mathbf{W}_\ell^O \text{Concat}[\mathbf{A}_\ell(\mathbf{h})] \quad (4)$$

Differentiable top- k masks. Having defined the pruning units, we associate each of them with a trainable mask logit. For a module at layer ℓ , let $\mathbf{z}_\ell$ denote its trainable mask logits. Given a pruning budget, we determine the number k of pruning units to retain and compute the continuous scores as follows:

$$\mathbf{s}_\ell = \sigma(\mathbf{z}_\ell/\tau) \quad (5)$$

where σ is the logistic sigmoid function and $\tau > 0$ is a temperature parameter controlling the sharpness of the scores. We then perform a top- k selection over $\mathbf{s}_\ell$ to construct a binary mask $\mathbf{b}_\ell$ that retains the k pruning units with the largest scores. Since this hard top- k selection is not differentiable, we use the straight-through estimator defined in Equation (6):

$$\tilde{\mathbf{m}}_\ell = \mathbf{b}_\ell + \mathbf{s}_\ell - \text{sg}(\mathbf{s}_\ell) \quad (6)$$

where sg denotes stop-gradient. Thus, the forward pass uses the hard mask $\mathbf{b}_\ell$, while gradients are propagated through the continuous scores $\mathbf{s}_\ell$. We

use $\tilde{\mathbf{m}}_\ell^{\text{ffn}}$ and $\tilde{\mathbf{m}}_\ell^{\text{gqa}}$ to denote the FFN- and GQA-specific masks, respectively.

For FFN channels, $\tilde{\mathbf{m}}_\ell^{\text{ffn}}$ is applied to the intermediate representation before the down projection, as follows:

$$\text{FFN}_\ell(\mathbf{h}; \tilde{\mathbf{m}}_\ell^{\text{ffn}}) = W_\ell^{\text{down}} (\tilde{\mathbf{m}}_\ell^{\text{ffn}} \odot \mathbf{u}_\ell(\mathbf{h})) \quad (7)$$

For GQA groups, $\tilde{\mathbf{m}}_\ell^{\text{gqa}}$ is applied to the output of the corresponding GQA pruning unit before the attention output projection:

$$\text{Attn}_\ell(\mathbf{h}; \tilde{\mathbf{m}}_\ell^{\text{gqa}}) = \mathbf{W}_\ell^O \text{Concat} [\tilde{\mathbf{m}}_\ell^{\text{gqa}} \odot \mathbf{A}_\ell(\mathbf{h})] \quad (8)$$

Together, the FFN and GQA masks determine which structured pruning units are retained in each layer for a given pruning budget.

Mask optimization. We optimize the mask logits on a task-specific machine translation calibration set while keeping the parameters of the base model frozen. Using the original unmasked model as the teacher, the optimization objective is defined as in Equation (9):

$$\mathcal{L} = \alpha \mathcal{L}_{\text{CE}} + (1 - \alpha) \mathcal{L}_{\text{KD}} + \beta \mathcal{L}_{\text{rep}} \quad (9)$$

where $\mathcal{L}_{\text{CE}}$ denotes the next-token cross-entropy between the masked model predictions and the reference translation tokens, while $\mathcal{L}_{\text{KD}}$ is the Kullback-Leibler divergence between the output distributions of the masked student and the unmasked teacher. Both terms are computed only at positions predicting tokens in the target translation. $\mathcal{L}_{\text{rep}}$ is the normalized squared L2 distance between the teacher and student outputs of each FFN and attention module, averaged across all modules.

During optimization, the pruning budget is introduced progressively and the mask temperature is annealed to encourage increasingly sharp selections. For configurations that prune both FFN and GQA component types, we manually set the number of GQA groups to retain to ensure that a minimum number of groups is preserved. The number of retained FFN channels is then adjusted to match the target parameter budget. Finally, the selected pruning units are fixed and physically removed.

3.2 Post-pruning Recovery and Complementary Compression

We explored three complementary approaches to recover performance and further reduce model size and memory requirements: supervised fine-tuning, vocabulary pruning, and AWQ quantization.

Supervised fine-tuning. We apply supervised fine-tuning to the structurally pruned model using task-specific machine translation data. The training loss is computed only over the target translation, excluding the instruction and source sentence.

Vocabulary pruning. We use the writing scripts associated with each language pair and only retain tokens whose alphabetic characters belong only to those scripts. Tokens containing digits, punctuation, whitespace, or other non-alphabetic symbols are preserved. We additionally maintain all special and tokens required by the chat template. The retained token IDs are remapped to a contiguous vocabulary, and the corresponding rows of the input embedding and output language-model head are pruned accordingly.

Quantization. We additionally apply 4-bit AWQ using task-specific machine translation examples for calibration, reducing model memory requirements while aiming to preserve translation quality.

4 Experimental Setup

4.1 Models

Following the requirements of the constrained track, we used Gemma 3 12B IT as our foundation model. Since the shared task exclusively evaluates text-based machine translation, we removed the vision-specific components and retained the language model as the starting point for all experiments. The resulting text-only model also served as the uncompressed baseline and as the teacher during mask optimization.

4.2 Corpora

Following the constrained track requirements, we sourced all training data from the WMT 2026 MT task data releases¹. Our data selection strategy leveraged the available parallel corpora for each language pair, sampling from diverse sources of varying quality and domains. We arbitrarily selected one million translation instruction pairs per language combination as a compromise between data coverage and reducing post-training computational time.

We provide a detailed breakdown of the original and sampled datasets in Table 1. Our train-

¹<https://www2.statmt.org/wmt26/mtdata/>

Dataset name	Total Size	Samples
CES-DEU		
Statmt-news_commentary-18.1	244,831	244,831
OPUS-neulab_tedtalks-v1	96,738	96,738
OPUS-ted2020-v1	153,227	153,227
OPUS-opensubtitles-v2024	36,408,370	168,401
OPUS-dgt-v4	3,048,670	168,401
OPUS-europarl-v8	568,589	168,402
<i>Total</i>		1,000,000
ENG-ARA		
OPUS-globalvoices-v2018q4	59,196	59,196
Statmt-news_commentary-18.1	193,671	193,671
Statmt-tedtalks-2_clean	341,887	149,426
OPUS-ted2020-v1	403,716	149,426
OPUS-qed-v2.0a	500,898	149,426
OPUS-opensubtitles-v2024	87,893,568	149,426
OPUS-multuion-v1	9,759,125	149,429
<i>Total</i>		1,000,000
ENG-ZHO		
Statmt-news_commentary-18.1	442,927	442,927
OPUS-ted2020-v1	16,209	16,209
OPUS-opensubtitles-v2016	9,304,777	180,288
ParaCrawl-paracrawl-1_bonus	14,170,585	180,288
OPUS-multuion-v1	9,564,315	180,288
<i>Total</i>		1,000,000

Table 1: Statistics of the training datasets used in our experiments. Total Size indicates the original dataset size, while Samples indicates the number of translation pairs selected for post-training.

ing data consisted of one million translation instruction pairs for each of the three language pairs (Czech–German, English–Simplified Chinese, and English–Egyptian Arabic), yielding a total of 3 million translation instructions. We detail the specific instruction template used for translation post-training in Appendix A. For mask optimization, we used 32,768 machine translation calibration examples sampled from the News Commentary data and distributed evenly across the three language pairs. For evaluation, we used the recommended development data for the WMT 2026 General MT Shared Task. Table 2 reports the number of translation pairs for each language pair.

4.3 Compression

Structured Pruning. We applied structured pruning to the text-only Gemma 3 12B IT model at target compression ratios of 0.25 and 0.50, where the compression ratio refers to the proportion of

Language Pair	Dataset	Samples
CES-DEU	newstest2019 (WMT 2024)	1,997
ENG-ARA	WMT24++ (Deutsch et al., 2025)	998
ENG-ZHO	WMT24++ (Deutsch et al., 2025)	998

Table 2: Development set statistics in terms of number of sentence pairs.

model parameters removed relative to the original model. We evaluated two configurations: pruning only FFN channels and jointly pruning FFN channels and GQA groups. In the joint configuration, we removed 25% of the GQA groups for the 0.25 compression setting and 50% for the 0.50 setting, retaining six and four groups, respectively. In both configurations, the number of retained FFN channels was automatically determined from the target global parameter budget; in the joint configuration, this calculation accounted for the manually specified number of retained GQA groups. We trained the masks for four epochs with a maximum sequence length of 512, a batch size of 2, and 32 gradient accumulation steps, resulting in an effective batch size of 64. We used a learning rate of 10^{-2} .

We assigned equal weight to the cross-entropy and distillation terms, using a distillation temperature of 1.0, and weighted intermediate representation matching by 0.02 to serve as an auxiliary regularization objective. Ablation results for the mask optimization objective are provided in Appendix B. These results show that knowledge distillation provides large improvements over cross-entropy alone, whereas the additional $\mathcal{L}_{\text{rep}}$ term has a comparatively smaller effect. The pruning ratio increased linearly from zero to the full target over the first 70% of optimizer updates in the first epoch and remained fixed thereafter. We also enforced a uniform structure across layers, using the same retained FFN width and the same number of retained GQA groups in every layer. FFN widths were further constrained to multiples of 128 to ensure compatibility with efficient inference engines and hardware-aligned tensor dimensions. The resulting parameter counts, FFN widths, and numbers of retained GQA groups are reported in Appendix B.

AWQ Quantization. We applied 4-bit AWQ quantization using llm-compressor (AI and vLLM Project, 2024). For calibration, we sampled 256 examples from each of the three development sets

Method	Compression	Size (GiB)	Time (s)	CES-DEU		ENG-ARA		ENG-ZHO	
				chrF	COMET	chrF	COMET	chrF	COMET
gemma-3-12b-it	-	21.9	131.19	54.9	0.8461	40.5	0.7781	35.8	0.8418
FFN	0.25	16.7	117.62	52.6	0.8349	39.6	0.7761	34.1	0.8298
FFN-GQA	0.25	16.7	120.13	52.1	0.8236	38.6	0.7630	32.1	0.8178
FFN	0.50	11.2	102.04	45.4	0.7132	34.4	0.7004	26.2	0.7518
FFN-GQA	0.50	11.2	98.82	39.0	0.6356	26.5	0.6337	18.4	0.6968

Table 3: Translation results for the baseline and structurally pruned models. Size denotes the checkpoint size, while Time reports the total inference time across all three language pairs.

listed in Table 2. We employed the asymmetric W4A16 scheme and targeted all linear layers except the output language-model head.

4.4 Post-training

We applied supervised fine-tuning to the pruned models using three million translation instructions. Training was performed for one epoch with a maximum sequence length of 2,048, sequence packing, and loss computed only over the assistant response. We used a batch size of 8, with four gradient accumulation steps, resulting in an effective batch size of 32. We used a learning rate of 10^{-5} , a cosine learning-rate schedule with a 3% warm-up ratio, and bfloat16 precision.

4.5 Inference

We used vLLM (Kwon et al., 2023) for efficient inference across all experiments. All translations were generated in a zero-shot setting using a fixed instruction prompt that requested only the target-language translation. Since VRAM usage was one of the main shared-task evaluation criteria, we explicitly configured vLLM’s `gpu_memory_utilization` rather than relying on its default behavior of reserving nearly all available GPU memory. For each model, we automatically estimated the required VRAM from its weights, numerical precision, KV cache, and runtime overhead, ensuring that the reported memory usage more accurately reflected the model’s inference requirements. As a conservative measure, we reserved sufficient VRAM to accommodate a maximum sequence length of 1,024 tokens for all translations, even though most examples are substantially shorter.

4.6 Evaluation

We evaluated translation quality using chrF (Popović, 2015) and COMET (Rei et al., 2020)

with the Unbabel/wmt22-comet-da model². We report checkpoint size as the primary model-size metric since VRAM consumption depends strongly on inference settings such as batch size and the memory headroom reserved by the inference configuration described above. Appendix C provides VRAM usage results as a factor of batch size, considering the aforementioned caveats.

For inference speed evaluation, we report timing measurements obtained using a batch size of 4,096 and bfloat16 precision for the non-quantized models. We excluded model loading time because of its high variability. Since the measurements were not averaged over repeated runs, they should be regarded as indicative estimates rather than precise performance figures; multiple runs would be necessary for a more reliable assessment.

4.7 Hardware

For the compression process and inference experiments, we used a single NVIDIA L40 GPU with 48GB of VRAM. For SFT post-training we used 4 NVIDIA H100 GPUs with 80GB of VRAM.

5 Results

We present our experimental results on the effects of structural pruning, supervised fine-tuning, quantization, and vocabulary pruning.

FFN pruning outperforms joint FFN-GQA pruning in terms of quality preservation. Table 3 compares the uncompressed baseline with models obtained using FFN-only and joint FFN-GQA pruning at compression ratios of 0.25 and 0.50. At 0.25 compression, FFN-only pruning reduces the checkpoint size from 21.9 to 16.7 GiB. Averaged across the three translation directions, chrF decreases by 1.6 points and COMET by 0.0084 relative to the baseline, indicating that

²<https://huggingface.co/Unbabel/wmt22-comet-da>

Method	Compression	CES-DEU		ENG-ARA		ENG-ZHO		Avg. Δ after SFT	
		chrF	COMET	chrF	COMET	chrF	COMET	chrF	COMET
FFN + SFT	0.25	53.5	0.8435	37.1	0.7542	27.8	0.7699	-2.63	-0.0244
FFN-GQA + SFT	0.25	53.6	0.8433	37.5	0.7579	27.2	0.7633	-1.50	-0.0133
FFN + SFT	0.50	52.1	0.8310	36.8	0.7452	25.7	0.7571	+2.87	+0.0560
FFN-GQA + SFT	0.50	52.0	0.8334	35.8	0.7471	24.3	0.7507	+9.40	+0.1217

Table 4: SFT results. The final two columns report the average absolute change across the three language pairs relative to the corresponding compressed model without SFT. Positive values indicate an improvement, whereas negative values indicate a performance degradation.

Method	Compression	Size (GiB)	Time (s)	CES-DEU		ENG-ARA		ENG-ZHO		Avg. Δ after AWQ	
				chrF	COMET	chrF	COMET	chrF	COMET	chrF	COMET
gemma-3-12b-it + AWQ	-	7.1	105.1	54.8	0.8445	40.0	0.7733	34.8	0.8387	-0.53	-0.0032
FFN + AWQ	0.25	5.7	92.9	52.4	0.8302	39.5	0.7742	33.7	0.8253	-0.23	-0.0037
FFN + AWQ	0.50	4.2	81.6	44.6	0.7012	33.7	0.6856	25.7	0.7454	-0.67	-0.0111
FFN + SFT + AWQ	0.50	4.2	80.0	50.9	0.8216	35.8	0.7388	23.5	0.7396	-1.47	-0.0111

Table 5: AWQ results. The final two columns report the average absolute change across the three language pairs relative to the corresponding model before AWQ quantization. Negative values indicate a performance drop.

most of the baseline translation quality is retained at this compression level. Increasing FFN-only compression to 0.50 reduces the checkpoint to 11.2 GiB. However, the average quality degradation increases to 8.4 chrF points and 0.1002 COMET under this more aggressive compression ratio. This result shows that the relationship between parameter reduction and translation quality is markedly nonlinear. We observe a general trend of faster inference as compression increases, as expected.

Under the same checkpoint-size budget, FFN-only pruning consistently outperforms joint FFN-GQA pruning. At 0.25 compression, joint pruning produces an additional average loss of 1.2 chrF points and 0.0121 COMET compared to FFN-only pruning. The disadvantage becomes substantially larger at 0.50 compression, where joint pruning incurs an additional average loss of 7.4 chrF points and 0.0664 COMET. These results suggest that GQA groups are more sensitive pruning units than FFN channels and that allocating the parameter-reduction budget primarily to FFN layers provides a more favorable quality-efficiency trade-off.

SFT is most effective under aggressive compression. Table 4 shows that the effect of SFT depends strongly on the compression ratio. At 0.25 compression, SFT decreases average chrF by 7.69% for FFN pruning and by 5.08% for joint FFN-GQA pruning. Average COMET likewise decreases by 3.00% and 1.65%, respectively.

Thus, for mildly compressed models, SFT did not improve translation quality and instead produced an overall degradation in our experiments.

At 0.50 compression, the trend reverses. SFT improves average chrF by 6.61% and COMET by 7.87% for FFN pruning. The improvements are even larger for joint FFN-GQA pruning, reaching 33.50% in chrF and 18.92% in COMET. These results indicate that SFT becomes increasingly beneficial as compression grows and the initial pruning-induced degradation becomes more severe. SFT is therefore particularly important for aggressively compressed models, where it can recover a substantial portion of the translation quality lost during pruning.

Since FFN-only pruning provided better overall translation performance, we used FFN-only pruning for the remaining experiments.

AWQ provides substantial size and speed gains with limited additional quality loss. Table 5 shows that applying AWQ reduces checkpoint size by between 62.5% and 67.6% relative to the corresponding full-precision models. This reduction introduces only modest additional degradation for most configurations. Across all models, the average chrF decrease ranges from 0.23 to 1.47 points, while the average COMET decrease ranges from 0.0032 to 0.0111. The 0.25-compressed FFN model is particularly robust to quantization, losing only 0.23 chrF points and 0.0037 COMET on average.

Method	Compression	CES-DEU			ENG-ARA			ENG-ZHO		
		Size (GiB)	chrF	COMET	Size (GiB)	chrF	COMET	Size (GiB)	chrF	COMET
gemma-3-12b-it + AWQ + VOCAB	–	6.5	54.8	0.8445	6.5	40.0	0.7733	6.6	34.8	0.8387
FFN + SFT + AWQ + VOCAB	0.50	3.6	50.9	0.8216	3.7	35.7	0.7388	3.8	23.5	0.7396

Table 6: Translation results for the AWQ-quantized baseline and the compressed, fine-tuned, and quantized model. Checkpoint size is reported separately for each language pair.

The SFT model is less stable under AWQ, with an average chrF decrease of 1.47 points, compared with 0.67 points for the corresponding FFN model without SFT. However, both models exhibit the same average COMET decrease of 0.0111. AWQ also consistently reduces inference time relative to the full-precision results in Table 3, following the general trend of faster inference after quantization. Overall, AWQ offers a favorable final compression step, providing large reductions in checkpoint size and faster inference while introducing limited additional quality loss.

Vocabulary pruning enables further near-lossless compression. Table 6 shows that language-specific vocabulary pruning further reduces checkpoint size without materially affecting translation quality on the evaluated language pairs. For the AWQ-quantized baseline, vocabulary pruning reduces the checkpoint from 7.1 GiB to 6.5–6.6 GiB, corresponding to an additional reduction of approximately 7.0–8.5%, while chrF and COMET remain unchanged. For the compressed FFN + SFT + AWQ model, the checkpoint decreases from 4.2 GiB to 3.6–3.8 GiB, an additional reduction of approximately 9.5–14.3%. Translation quality is again essentially preserved, with identical COMET scores and only a 0.1 point chrF decrease for ENG–ARA, which may be attributable to numerical precision effects.

These results show that removing vocabulary entries not required by a specific language pair can provide further near-lossless compression. However, this conclusion assumes that the retained vocabulary adequately covers the deployment inputs. For unrestricted or open-domain text, previously unseen words, scripts, or symbols may rely on pruned vocabulary entries, potentially degrading translation quality. Statistics on the retained vocabulary sizes and corresponding parameter counts after vocabulary pruning are provided in Appendix D.

6 Submissions

Our primary submission for the task was a multilingual model compressed at 0.50 under the FFN+SFT+AWQ setting, as FFN-only pruning provided the most stable results across compression levels and language pairs, with SFT being able to provide significant quality recovery at 0.50 compression, and AWQ providing further compression and inference gains with negligible additional quality loss. We also submitted our other variants as contrastive runs, namely the FFN-pruned models at 0.25 and 0.50 compression with 4-bit AWQ, prior to SFT. Additionally, we submitted language-specific variants of our primary FFN+SFT+AWQ configuration with vocabulary pruning for each individual translation pair.

7 Conclusions

In this work, we presented Vicomtech’s approach to the constrained track of the WMT 2026 Model Compression shared task. We compressed the text-only Gemma 3 12B IT model for Czech-to-German, English-to-Simplified-Chinese, and English-to-Egyptian-Arabic. Our method learns differentiable masks over FFN channels and GQA groups from machine translation calibration data, combining cross-entropy and knowledge distillation. We complemented structured pruning with supervised fine-tuning, 4-bit AWQ quantization, and language-specific vocabulary pruning.

Our experiments showed that task-aware structured pruning can successfully reduce model size while preserving competitive translation quality, particularly at moderate compression levels. Among the evaluated variants, FFN channels proved to be a more reliable pruning target than GQA groups, consistently providing a better quality–efficiency trade-off under the same parameter budget. More aggressive compression remained feasible, although it introduced larger performance losses and required post-training to recover part of the degraded translation quality.

AWQ provided the largest additional reduc-

tion in checkpoint size, with comparatively limited quality degradation, while vocabulary pruning yielded an additional near-lossless reduction when the expected source–target languages were known in advance. The complete pipeline reduced the original 21.9 GiB checkpoint to 3.6–3.8 GiB, corresponding to an overall size reduction of approximately 83%. Overall, our findings establish structured pruning as a flexible foundation for deploying multilingual translation models in resource-constrained environments.

References

- Red Hat AI and vLLM Project. 2024. [LLM Compressor](#).
- Duarte Miguel Alves, José Pombal, Nuno M Guerreiro, Pedro Henrique Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and Andre Martins. 2024. [Tower: An open multilingual large language model for translation-related tasks](#). In *First Conference on Language Modeling*.
- Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. 2024. [Fluctuation-based adaptive structured pruning for large language models](#). In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence, AAAI’24/IAAI’24/EAAI’24*. AAAI Press.
- Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoeffler, and James Hensman. 2024. SliceGPT: Compress large language models by deleting rows and columns. *arXiv preprint arXiv:2401.15024*.
- Tom Brown et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.
- Yupeng Chang et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.
- John Dang, Shivalika Singh, Daniel D’souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, Acyr Locatelli, Hangyu Lin, Dwarak Talupuru, Bharat Venkitesh, David Cairuz, Bowen Yang, Tim Chung, Wei-Yin Ko, Sylvie Shang Shi, Amir Shukayev, Sammie Bae, Aleksandra Piktus, Roman Castagné, Felipe Cruz-Salinas, Eddie Kim, Lucas Crawhall-Stein, Adrien Morisot, Sudip Roy, Phil Blunsom, Ivan Zhang, Aidan Gomez, Nick Frosst, Marzieh Fadaee, Beyza Ermis, Ahmet Üstün, and Sara Hooker. 2024. [Aya expande: Combining research breakthroughs for a new multilingual frontier](#).
- Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoeffler, and Dan Alistarh. 2023. SpQR: A sparse-quantized representation for near-lossless LLM weight compression. *arXiv preprint arXiv:2306.03078*.
- Daniel Deutsch, Eleftheria Briakou, Isaac Rayburn Caswell, Mara Finkelstein, Rebecca Galar, Juraj Juraska, Geza Kovacs, Alison Lui, Ricardo Rei, Jason Riesa, Shruti Rijhwani, Parker Riley, Elizabeth Salesky, Firas Trabelsi, Stephanie Winkler, Biao Zhang, and Markus Freitag. 2025. [WMT24++: Expanding the language coverage of WMT24 to 55 languages & dialects](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12257–12284, Vienna, Austria. Association for Computational Linguistics.
- Gemma Team. 2025. [Gemma 3 technical report](#). *arXiv preprint arXiv:2503.19786*.
- 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.
- Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz,

- Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, Mariya Shmatova, Steinhórfur Steingrímsson, and Vilém Zouhar. 2024. [Findings of the WMT24 general machine translation shared task: The LLM era is here but MT is not solved yet](#). In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46, Miami, Florida, USA. Association for Computational Linguistics.
- Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Masaaki Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, Mariya Shmatova, and Jun Suzuki. 2023. [Findings of the 2023 conference on machine translation \(WMT23\): LLMs are here but not quite there yet](#). In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.
- 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*, volume 6, pages 87–100.
- Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. LLM-pruner: On the structural pruning of large language models. *Advances in Neural Information Processing Systems*, 36:21702–21720.
- Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M Guerreiro, Ricardo Rei, Duarte M Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, et al. 2025. Eurollm: Multilingual language models for europe. *Procedia Computer Science*, 255:53–62.
- David Ponce, Thierry Etchegoyhen, and Javier Del Ser. 2025a. Gelaco: An evolutionary approach to layer compression. *arXiv preprint arXiv:2507.10059*.
- David Ponce, Harritxu Gete, and Thierry Etchegoyhen. 2025b. [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.
- 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.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).
- 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.
- Oliver Sieberling, Denis Kuznedelev, Eldar Kurtic, and Dan Alistarh. 2024. EvoPress: Towards optimal dynamic model compression via evolutionary search. *arXiv preprint arXiv:2410.14649*.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. [Energy and policy considerations for deep learning in NLP](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.
- Ziyang Wang, Jiangfeng Xiao, Chuan Xiao, Ruoxiang LI, Rui Mao, and Jianbin Qin. 2026.

GRASPrune: Global gating for budgeted structured pruning of large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10719–10736, San Diego, California, United States. Association for Computational Linguistics.

Edward WD Whittaker and Bhiksha Raj. 2001. Quantization-based language model compression. In *INTERSPEECH*, pages 33–36.

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 *International Conference on Machine Learning*, pages 38087–38099. PMLR.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024. [A paradigm shift in machine translation: Boosting translation performance of large language models](#). In *The Twelfth International Conference on Learning Representations*.

Yifei Yang, Zouying Cao, and Hai Zhao. 2024. [LaCo: Large language model pruning via layer collapse](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6401–6417, Miami, Florida, USA. Association for Computational Linguistics.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. [Multilingual machine translation with large language models: Empirical results and analysis](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781, Mexico City, Mexico. Association for Computational Linguistics.

A Translation Instruction Template

The following template illustrates the format used for translation instructions. At inference time, the system and user messages are provided to the model, followed by the generation prompt defined by the model’s chat template. The instruction prompt and language names are always specified in English. In the template, SOURCE_LANGUAGE and TARGET_LANGUAGE represent the English names of the source and target languages (e.g., "Czech", "German", "English", "Egyptian Arabic" or "Simplified Chinese"), and INPUT_SENTENCE contains the text to be translated in the source language.

Instruction Template

```
"messages": [
  {
    "role": "system",
    "content": "You are a helpful and precise translator that translates from
SOURCE_LANGUAGE to TARGET_LANGUAGE. Please provide only the translation without
any additional explanations or formatting."
  },
  {
    "role": "user",
    "content": "Translate from SOURCE_LANGUAGE to TARGET_LANGUAGE:\nINPUT_SENTENCE"
  }
]
```

B Pruned Model Architectures

Table 7 summarizes the parameter counts and architectural dimensions of the original Gemma 3 12B IT model and its compressed variants.

Model	# Parameters	FFN channels	GQA groups
Gemma-3-12B-IT	11,766,034,176	15,360	8
FFN 0.25	8,864,100,096	10,112	8
FFN-GQA 0.25	8,793,321,216	11,008	6
FFN 0.50	5,891,387,136	4,736	8
FFN-GQA 0.50	5,891,387,136	6,784	4

Table 7: Architectural characteristics of the original and compressed models.

C VRAM Usage

Figure 2 reports the estimated VRAM usage as a function of inference batch size for our primary submission. VRAM usage increases with batch size and reaches the configured GPU memory limit at approximately batch size 128.

D Vocabulary Pruning Statistics

Model	Vocabulary size	Embedding / LM head parameters
Original	262,208	1,006,878,720
CES-DEU	167,942	644,897,280
ENG-ARA	176,267	676,865,280
ENG-ZHO	187,198	718,840,320

Table 8: Vocabulary size and corresponding embedding/LM-head parameter count for the original and language-specific vocabulary-pruned models.

E Mask Optimization Objective Ablation

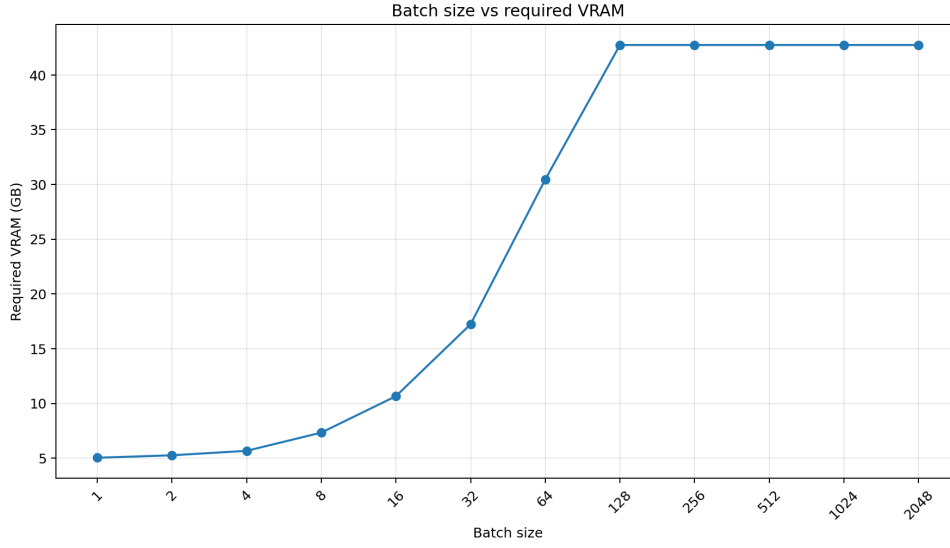


Figure 2: Estimated VRAM usage as a function of inference batch size for our primary submission.

Method	Objective	CES-DEU		ENG-ARA		ENG-ZHO	
		chrF	COMET	chrF	COMET	chrF	COMET
FFN 0.25	$\mathcal{L}_{CE}$	39.4	0.7401	27.5	0.6265	21.2	0.6734
	$\mathcal{L}_{CE} + \mathcal{L}_{KD}$	52.9	0.8350	39.2	0.7685	33.6	0.8237
	$\mathcal{L}_{CE} + \mathcal{L}_{KD} + \mathcal{L}_{rep}$	52.6	0.8349	39.6	0.7761	34.1	0.8298
FFN-GQA 0.25	$\mathcal{L}_{CE}$	27.1	0.5284	25.5	0.5686	17.3	0.5264
	$\mathcal{L}_{CE} + \mathcal{L}_{KD}$	52.1	0.8304	38.7	0.7659	33.0	0.8255
	$\mathcal{L}_{CE} + \mathcal{L}_{KD} + \mathcal{L}_{rep}$	52.1	0.8236	38.6	0.7630	32.1	0.8178
FFN 0.50	$\mathcal{L}_{CE}$	21.9	0.3987	21.8	0.4863	13.7	0.4866
	$\mathcal{L}_{CE} + \mathcal{L}_{KD}$	44.9	0.7250	33.5	0.6887	26.0	0.7527
	$\mathcal{L}_{CE} + \mathcal{L}_{KD} + \mathcal{L}_{rep}$	45.4	0.7132	34.4	0.7004	26.2	0.7518
FFN-GQA 0.50	$\mathcal{L}_{CE}$	20.3	0.3605	16.0	0.4385	8.1	0.4008
	$\mathcal{L}_{CE} + \mathcal{L}_{KD}$	37.5	0.6107	25.8	0.6254	16.6	0.6686
	$\mathcal{L}_{CE} + \mathcal{L}_{KD} + \mathcal{L}_{rep}$	39.0	0.6356	26.5	0.6337	18.4	0.6968

Table 9: Ablation of the mask-optimization objective for FFN-only and joint FFN-GQA structured pruning at compression ratios of 0.25 and 0.50.