COMPRESS LOCALLY, PREDICT GLOBALLY: COMPOSABLE MARGIN STATISTICS FOR ESTIMATING COMPRESSED LLM ACCURACY

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce a statistical model that predicts the task accuracy of compressed language models without constructing and evaluating each candidate configuration. For each block and compression option, we summarize changes in answer margins (the score differences between competing answers) by a scaling coefficient and a residual variance. We estimate these statistics by compressing one block at a time, then reuse and combine them across configurations. We use the combined statistics to parameterize a Gaussian score model. For each example in a labeled reference set, we use the uncompressed model's scores to estimate the probability that the compressed model selects the correct answer. We average these probabilities to predict task accuracy. Across exhaustively evaluated MMLU quantization and pruning allocation spaces, prediction mean absolute error (MAE) ranges from 0.61 to 3.83 percentage points (pp). Across five K2-Horizon-7B quantization spaces, composing margin statistics achieves a mean per-space prediction MAE of 1.10 pp, compared with 1.55 and 1.42 pp for additive and multiplicative accuracy composition. Comparisons with covariance-based and empirical-residual predictors further support the accuracy estimates from the fitted Gaussian model. GPTQ prediction MAE remains within 0.67–1.19 pp across five checkpoints from 7B to 65B and reaches 0.90 pp on 24 presampled individual-layer allocations. The same prediction rule extends to BoolQ, HellaSwag, WinoGrande, and encoder-based AG News classification. As an application, maximizing predicted accuracy selects the best measured configuration in the complete Gemma-31B quantization space and improves pruning allocation over OWL under matched budgets.

1 INTRODUCTION

Quantization and pruning reduce the storage required for language-model weights (Frantar et al., 2023; Frantar & Alistarh, 2023; Sun et al., 2024; Lin et al., 2024). For a fixed compression procedure and resource budget, different allocations of weight precision or sparsity across layers can yield different task accuracies. The number of feasible configurations grows rapidly: assigning two precision levels to 36 layers already yields billions of allocations at a fixed mean bit width. Directly evaluating these choices requires running each complete compressed model. Predicting their accuracy from reusable local measurements would allow configurations to be compared and selected before each candidate is constructed and evaluated.

Existing approaches use reconstruction error and quantization sensitivity to assess local compression effects (Frantar et al., 2023; Zandonati et al., 2023). Other approaches predict perplexity changes (Malinovskii et al., 2025) or evaluate compressed candidates to search for an allocation (Sieberling et al., 2025). Reconstruction error, sensitivity, and perplexity do not directly specify task accuracy, while candidate-based search requires model execution for the configurations it examines.

We approach task-accuracy prediction through answer margins, the score differences between competing answers. The model answers correctly when the correct answer scores above every alternative. The effect of a score perturbation therefore depends on the original margins: a change that reverses a narrow lead can leave a wider lead intact. Compressing a single block can change these

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

margins without changing the selected answer. A single-block accuracy measurement does not record such changes, even though they can affect the outcome of further compression.

We compose margin statistics before converting them to predicted task accuracy. We partition the transformer into blocks of consecutive layers, compress one block at a time, and measure its effect on answer margins with the remaining blocks uncompressed. For each block and compression option, a scaling coefficient describes systematic margin rescaling and a residual variance describes the remaining variation. We use the combined statistics and dense-model scores to estimate correct-answer probabilities under a Gaussian score model. Averaging these probabilities over a labeled reference set gives predicted task accuracy. The resulting parameter library grows with the number of block options and is reused across configurations, allowing additional candidates to be assessed without compressed-model execution. Figure 1 illustrates the margin intuition and prediction procedure.

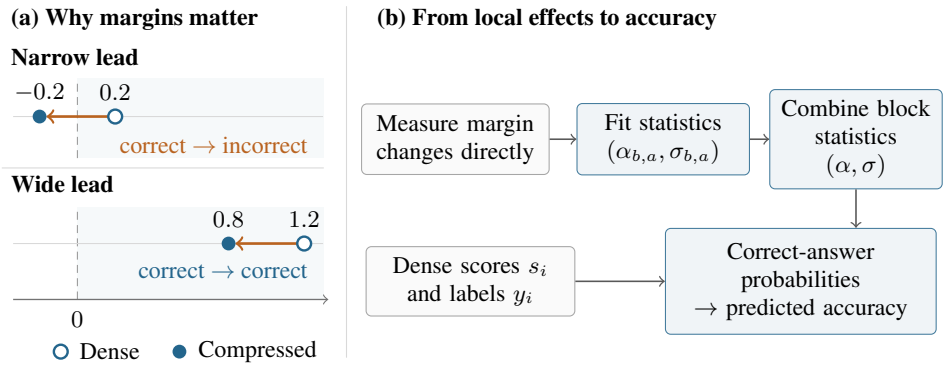

Figure 1: Margin-based accuracy prediction. Panel (a) shows schematic two-choice answer margins under the same perturbation; zero marks the decision boundary.

We first test whether these local statistics predict accuracy throughout complete MMLU quantization and pruning allocation spaces, achieving 0.61–3.83 percentage points (pp) of mean absolute error (MAE) in accuracy prediction. Across five K2-Horizon-7B quantization spaces, composing margin statistics gives 1.10 pp mean per-space prediction MAE, compared with 1.55 and 1.42 pp for additive and multiplicative accuracy composition. The fitted Gaussian predictor also yields lower prediction MAE than covariance-based and empirical-residual alternatives across six quantization spaces. Additional experiments extend prediction across tasks, model scales from 7B to 65B (0.67–1.19 pp prediction MAE for GPTQ), and individual-layer combinations, with encoder-based news classification extending the score model beyond decoder LLMs. Precision, sparsity, and retrospective multitask allocation illustrate how these predictions guide compression decisions. We make three contributions:

- A statistical model that connects compression-induced changes in answer margins to correct-answer probabilities and quantitative task-accuracy predictions.
- A blockwise composition method that reuses locally measured margin scaling and residual variance to predict complete-configuration accuracy without evaluating each candidate model.
- Empirical validation of the margin model for accuracy prediction and its application to task-aware compression allocation.

2 Related Work

Post-training compression and allocation. Wanda (Sun et al., 2024) scores weights using weight magnitudes and input activation norms, while SparseGPT (Frantar & Alistarh, 2023) combines pruning with weight updates to reduce layer-output reconstruction error. GPTQ (Frantar et al., 2023) uses approximate second-order information for weight quantization and error compensation. AWQ (Lin et al., 2024) uses activation-aware per-input-channel scaling and weight clipping. OWL (Yin et al.,

2024) assigns layerwise sparsity using outlier ratios of activation-weighted weight scores. HAWQ and HAWQ-V2 (Dong et al., 2019; 2020) use curvature to allocate precision, and SliM-LLM (Huang et al., 2025) performs salience-driven mixed precision at the weight-group level.

Predicting the effect of quantization. FIT (Zandonati et al., 2023) develops a Fisher-based measure of model sensitivity to quantization. HIGGS (Malinovskii et al., 2025) relates relative squared weight error to expected perplexity increase through a linearity theorem and uses this relation for quantization and allocation. These works connect local quantization effects to model-level performance. We connect compression-induced answer-score changes to task accuracy.

Search over compression configurations. EvoPress (Sieberling et al., 2025) searches for a budget-feasible allocation using evolutionary mutations and evaluations of compressed candidates. Its motivating experiments show that lower sums of local error scores need not imply better compressed-model performance. Our contribution is an explicit accuracy predictor: cached block statistics estimate the task accuracy of combinations before candidate evaluation and provide an objective for allocation.

3 METHOD

3.1 PROBLEM FORMULATION

Consider an uncompressed (dense) transformer partitioned into L compression blocks, each containing one or more consecutive layers. Each block b has a finite set of compression options $\mathcal{A}_b$. A configuration $\boldsymbol{a} = (a_1, \ldots, a_L)$ assigns an option $a_b \in \mathcal{A}_b$ to each block, and $\mathcal{F}_B$ is the set of configurations satisfying resource budget B.

Let $\mathcal{D} = \{(s_i, y_i)\}$ denote a labeled reference set represented by dense-model answer scores. For example i, the vector $s_i \in \mathbb{R}^{K_i}$ contains scores for K_i candidate answers, and $y_i \in \{1, \ldots, K_i\}$ indexes the correct answer. The component s_{ic} is the score for candidate c, such as a continuation log-likelihood or a class logit. Our goal is to predict $A_{\mathcal{D}}(\boldsymbol{a})$, the accuracy of configuration $\boldsymbol{a}$ on these reference examples, for every $\boldsymbol{a} \in \mathcal{F}_B$ using local compression measurements, without evaluating each complete compressed model.

3.2 FROM ANSWER MARGINS TO ACCURACY

The correct answer y_i wins when all margins $s_{iy_i} - s_{ic}$, $c \neq y_i$, are positive. We represent compressed scores as rescaled dense scores plus additive residuals (Tse & Viswanath, 2005). The scaling captures systematic changes in margin magnitude; the residual distribution describes the remaining variation across examples and answer choices. Positive scaling preserves answer ordering, so answer changes depend on residual variation relative to the rescaled margins.

We model the compressed answer score $\widetilde{s}_{ic}$ using a scaling coefficient α and independent and identically distributed (iid) Gaussian score residuals:

$$\widetilde{s}_{ic} = \alpha s_{ic} + \epsilon_{ic}, \qquad \epsilon_{ic} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2/2). \tag{1}$$

We omit score offsets shared by all choices because they leave the selected answer unchanged. The factor $1/2$ gives $\mathrm{Var}(\epsilon_{ic} - \epsilon_{ic'}) = \sigma^2$ for distinct choices c and c', so σ is the standard deviation of a pairwise margin residual. We estimate α without constraining it to $[0, 1]$, allowing both contraction and expansion of margins.

Accuracy prediction. Given α, σ, and the labeled reference set $\mathcal{D}$, we predict accuracy as

$$\widehat{A}_{\mathcal{D}}(\alpha, \sigma) = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \Pr\left[\arg\max_c \{\alpha s_{ic} + \epsilon_{ic}\} = y_i\right] \tag{2}$$

by computing correct-answer probabilities with Monte Carlo sampling over all answer choices.

For two choices, let $\bar{y}_i$ index the alternative to the correct answer y_i, and define their margin as $d_i = s_{iy_i} - s_{i\bar{y}_i}$. The probability of a correct answer is $\Phi(\alpha d_i/\sigma)$, where Φ is the standard normal

cumulative distribution function. With several choices, the identity of the strongest rival can change under perturbation. Simulating all choices retains this competition. For $\alpha > 0$, decision probabilities depend on the ratio σ/α; retaining the two parameters separately is useful when composing compression effects across blocks.

3.3 Estimating blockwise margin statistics

For each block b and option a, we compress only that block, keep the remaining blocks dense, and measure the resulting answer margins on the fitting examples. These measurements supply $(\alpha_{b,a}, \sigma_{b,a})$, with separate parameters for each model and task. We fit these parameters using the margin between the dense model's two highest-scoring answers, which requires no correct-answer labels. Suppressing the block and option indices, let p_i and q_i index the dense model's highest- and second-highest-scoring answers, respectively, and write their dense margin as $m_i = s_{ip_i} - s_{iq_i}$. Holding these choice identities fixed, compression changes the margin to $m_i + \Delta m_i$. A least-squares regression through the origin over n fitting examples gives

$$\widehat{\alpha} = 1 + \frac{\sum_i m_i \Delta m_i}{\sum_i m_i^2}, \tag{3}$$

$$r_i = \Delta m_i - (\widehat{\alpha} - 1) m_i, \qquad \widehat{\sigma}^2 = \frac{1}{n} \sum_i (r_i - \bar{r})^2. \tag{4}$$

Here $\bar{r} = n^{-1} \sum_i r_i$ is the sample mean of the margin residuals. Because the choice identities remain fixed, the compressed margin becomes negative when their order reverses. Equation 2 then applies these parameters to all answer choices and uses the reference labels to predict accuracy. Measuring each block option once supplies a parameter library that is reused across allocations.

3.4 Composing compression effects across blocks

Having estimated each block's margin statistics, we combine them to predict the effect of a complete configuration. To motivate the composition rule, consider a scalar model in which v_0 is a dense answer margin and each step applies $v_b = \alpha_b v_{b-1} + \eta_b$, with residual variance $\mathrm{Var}(\eta_b) = \sigma_b^2$. Each step rescales both the margin and the perturbations accumulated at earlier steps. Repeated substitution therefore motivates a product of scaling coefficients and a sum of residual variances weighted by later coefficients:

$$\alpha(\boldsymbol{a}) = \prod_{b=1}^{L} \alpha_{b,a_b}, \qquad \sigma^2(\boldsymbol{a}) = \sum_{b=1}^{L} \sigma_{b,a_b}^2 \left(\prod_{j>b} \alpha_{j,a_j} \right)^2. \tag{5}$$

Equation 5 is exact for the scalar recurrence with uncorrelated zero-mean residuals (Appendix A). We use it as an empirical composition model for separately measured transformer blocks. Substituting the combined parameters into Equation 2 gives the configuration's predicted accuracy from the cached block statistics and labeled reference set.

3.5 Budgeted allocation

Let $\mathcal{D}_{\text{select}}$ denote the labeled reference set reserved for configuration selection. Predicted accuracy supplies the objective:

$$\widehat{\boldsymbol{a}} = \arg\max_{\boldsymbol{a} \in \mathcal{F}_B} \widehat{A}_{\mathcal{D}_{\text{select}}} \big(\alpha(\boldsymbol{a}), \sigma(\boldsymbol{a}) \big). \tag{6}$$

For quantization, we constrain the average bit width, computed by weighting each block's bit width by its number of quantized weights. For pruning, we constrain the fraction of targeted weights removed. Small configuration spaces can be enumerated; larger spaces can be searched without constructing each candidate model. For multiple tasks, the objective is a weighted mean of predicted task accuracies, using separately estimated block statistics for each task.

4 Experiments

We first test complete-space accuracy prediction across model families and compression methods, then examine the margin representation and Gaussian residual model. Task, model-size, and

individual-layer experiments test the breadth of the construction. Finally, we use its predictions to choose compression allocations.

4.1 Evaluation protocol

Models, compression settings, and reference data. Our main LLM evaluation uses six decoder checkpoints spanning 7B–65B: K2-Horizon-7B and 32B (IFM Team, 2026), Vicuna-7B v1.5 (Zheng et al., 2023), Gemma-4-12B and 31B pretrained (PT) models (Gemma Team, 2026), and LLM360 K2-65B (Liu et al., 2025). We shorten K2-Horizon to K2 and Gemma-4 to Gemma in size-qualified labels. We evaluate weight-only quantization with GPTQ, AWQ, and Hadamard-rotated GPTQ, and unstructured weight pruning with Wanda; allocations distribute 3-bit (W3) and 4-bit (W4) weights or sparsity levels across blocks. The model-family and compression-method comparisons in Table 1 use direct block measurements and the composition rule throughout.

MMLU (Hendrycks et al., 2021) uses five-shot prompts and 40 items per subject with lm-evaluation-harness 0.4.13 (Gao et al., 2024). A fixed subject-stratified split assigns 570 items to fitting, 570 to selection, and 1,140 to testing, with continuation log-likelihoods as answer scores. We fit block parameters from score changes on the fit split without using correct-answer labels, then select allocations by maximizing predicted accuracy on the selection reference. On the test split, predicted accuracies use the dense model's scores and correct-answer labels, and are compared with accuracies measured from compressed-model responses on the same items. Across allocations, we report the mean absolute error (MAE) of these accuracy predictions in percentage points (pp) and their Spearman rank correlation ρ with measured accuracy. In tables, gray shading identifies our method's results; blue marks the best comparable result within each row, including ties at the reported precision.

4.2 Accuracy prediction across model families

We first test whether reusable block measurements predict complete-model accuracy across model families under a common quantization setting. For K2-Horizon-7B, Vicuna-7B, Gemma-4-31B PT, and K2-Horizon-32B, we construct six-block GPTQ allocation spaces using W3 and W4 at a nominal average weight bit width of 3.50 bits. K2-Horizon-7B and Gemma-4-31B use equal-length blocks of six and ten layers, respectively; for each model, we enumerate all 20 assignments with three W4 blocks and three W3 blocks. Vicuna-7B's blocks contain 6/5/5/5/5/6 layers, yielding 12 allocations with W4 on exactly 16 of its 32 layers. For K2-Horizon-32B, we likewise enumerate all 20 assignments with three W4 blocks and three W3 blocks; unequal block sizes give average bit widths of 3.48–3.52.

For each model, 12 direct W3/W4 block measurements supply the statistics reused across all allocations. We use GPTQ (Frantar et al., 2023) with quantization groups of 128 weights and up to 128 C4 calibration sequences of length at most 2,048. Additional precision budgets and computational costs appear in Appendices G and F. The model-family comparison in Table 1 summarizes prediction results for all 72 allocations.

Across these four checkpoints, the predictor achieves 0.67–1.19 pp MAE in predicted accuracy using the same composition and probability equations.

Table 1: Complete-space MMLU accuracy prediction. N: number of allocations.

GPTQ, nominal 3.50 bits			
Model	N	MAE (pp) ↓	ρ ↑
K2-Horizon-7B	20	1.19	0.877
Vicuna-7B	12	0.67	0.867
Gemma-4-31B PT	20	1.04	0.777
K2-Horizon-32B	20	0.76	0.674

K2-Horizon-7B			
Method / budget	N	MAE (pp) ↓	ρ ↑
GPTQ, 3.50 bits	20	1.19	0.877
AWQ, 3.50 bits	20	0.97	0.922
Rotated GPTQ, 3.50 bits	20	0.61	0.820
Wanda, 60% sparsity	21	1.95	0.865

4.3 A SHARED PREDICTION RULE ACROSS COMPRESSION METHODS

We next test whether the margin model predicts accuracy across compression methods while holding the language model fixed. On K2-Horizon-7B, AWQ and Hadamard-rotated GPTQ each supply 12 W3/W4 parameter pairs for the same 20-allocation, 3.50-bit space used for GPTQ. Each method supplies its own block statistics, which the shared composition and probability equations map to task accuracy. The resulting prediction MAEs are 0.97 pp for AWQ and 0.61 pp for rotated GPTQ.

To extend this comparison from precision reduction to weight removal, we apply the Wanda pruning metric on K2-Horizon-7B using calibration activations from the dense model. Weight magnitudes and input-activation norms determine the pruning scores. The 60%-sparsity space contains the 20 assignments with three blocks at 50% and three at 70%, together with uniform 60% pruning. Across these allocations, measured accuracy spans 29.7 pp, and the predictor achieves 1.95 pp MAE (Figure 2).

Together, the two comparisons in Table 1 cover seven distinct allocation spaces and 133 configurations. Across GPTQ, AWQ, rotated GPTQ, and Wanda on K2-Horizon-7B, per-space MAE in predicted accuracy ranges from 0.61 to 1.95 pp under the same composition and probability equations.

We also evaluate a complete Wanda space on Gemma-12B PT at 55% sparsity. Across its 28 allocations, the same predictor achieves 3.83 pp prediction MAE and $\rho = 0.956$ (Appendix H).

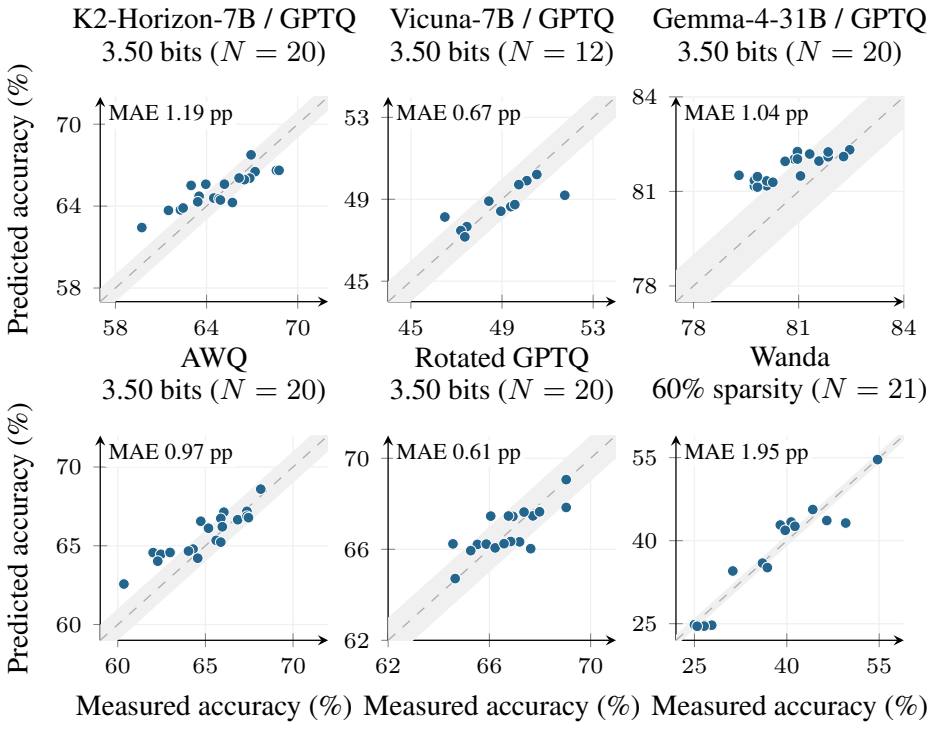

Figure 2: Complete-space MMLU accuracy prediction. Bottom row: K2-Horizon-7B. Dashed lines: equality; gray bands: ± 1 pp.

4.4 VALIDATING THE COMPRESSION ERROR MODEL

What information should be composed? We assess the value of composing margin statistics by comparing our predictor with two baselines that compose single-block accuracies. Table 2 compares five complete K2-Horizon-7B quantization spaces covering 90 allocations. Additive accuracy prediction sums each single-block accuracy change, while multiplicative accuracy prediction multiplies the ratios of single-block compressed accuracy to dense-model accuracy. Both baselines use selection-split single-block measurements and dense test accuracy as the reference level. All three

predictors use direct single-block measurements. Across the five spaces, composing margin statistics gives 1.10 pp unweighted mean per-space prediction MAE, compared with 1.55 pp for additive and 1.42 pp for multiplicative composition.

Table 2: K2-Horizon-7B MMLU accuracy-prediction MAE (pp).

Method / budget	Additive	Multiplicative	Ours
GPTQ, 3.33 bits	1.60	1.55	1.81
GPTQ, 3.50 bits	1.49	1.24	1.19
GPTQ, 3.67 bits	1.66	1.49	0.91
AWQ, 3.50 bits	1.60	1.59	0.97
Rotated GPTQ, 3.50 bits	1.42	1.20	0.61
Mean across spaces	1.55	1.42	1.10

Evaluating the Gaussian approximation. We examine the zero-mean residual approximation by measuring the mean of the fitted top-two margin residuals relative to their standard deviation. Across 448 measurement sets covering block options on seven tasks, the ratio of the absolute residual mean to its standard deviation has a median of 0.028 and a 95th percentile of 0.094, indicating that the residual mean is small relative to its variation.

We next test how the residual model affects accuracy prediction. We compare our fitted independent Gaussian predictor with two alternatives: a Gaussian model with a covariance matrix estimated using shrinkage and an empirical-residual bootstrap. The independent model estimates its scale from top-two margin residuals; the alternatives use full answer-score residual vectors to estimate covariance or resample residuals. This comparison of fitted predictors uses the same direct block measurements, combined scaling coefficients, and 96 Monte Carlo draws per item in a separate diagnostic rerun across six complete K2-Horizon-7B and Vicuna-7B quantization spaces. Appendix D reports the per-space results and item-level probability calibration. On these 102 allocations, the independent Gaussian predictor achieves lower prediction MAE than both alternatives in every space; its unweighted mean per-space prediction MAE is 1.03 pp, versus 1.38 pp for each alternative.

4.5 Extending accuracy prediction across tasks

Extending prediction to other multiple-choice tasks. To test whether accuracy prediction extends beyond MMLU, we evaluate tasks with different answer formats and scoring conventions. We apply the same composition and probability equations to all 20 K2-7B GPTQ 3.50-bit allocations on zero-shot BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), and WinoGrande (Sakaguchi et al., 2020), using task-specific reference scores, direct measurements, and fitted margin statistics. A fixed 25/25/50 fit/selection/test split gives test sets of 1,636, 1,000, and 635 items, respectively. BoolQ and WinoGrande use two answer choices and log-likelihoods without length normalization; HellaSwag uses four continuations and byte-normalized scores. On BoolQ, HellaSwag, and WinoGrande, prediction MAEs are 0.71, 1.40, and 1.17 pp, with $\rho = 0.821$, 0.760, and 0.632, respectively.

From decoder scores to encoder classification. To test the prediction rule on encoder classification, we use class logits from Laya, a 421M-parameter ModernBERT-based model (Convai Innovations, 2026; Warner et al., 2025), as answer scores on four-class AG News (Zhang et al., 2015). We evaluate four complete GPTQ or Wanda spaces of 35 allocations each, together with 100 individual-layer allocations sampled in advance. The same composition and probability equations give 0.46–1.85 pp prediction MAE across the blockwise spaces and 0.45 pp at individual-layer granularity (Table 3).

4.6 Scaling to larger language models

To examine accuracy prediction at larger model sizes, we extend the six-block GPTQ comparison to K2-65B, covering five decoder checkpoints from 7B to 65B with direct measurements and the composition rule at nominal 3.5 bits. The four smaller-model spaces are evaluated exhaustively; for K2-65B, we evaluate the first 16 of 20 assignments in the prescribed run order. Unequal block sizes

Table 3: AG News accuracy prediction with Laya on 1,000 test items. N: number of allocations; Wanda budgets denote sparsity.

Compression	Budget	N	MAE (pp) ↓	ρ ↑
GPTQ, four-layer blocks	3.43 bits	35	0.54	0.457
GPTQ, four-layer blocks	3.57 bits	35	0.46	0.608
Wanda, four-layer blocks	58.57%	35	1.67	0.946
Wanda, four-layer blocks	61.43%	35	1.85	0.888
GPTQ, individual layers	3.50 bits	100	0.45	0.406

produce small differences in average bit width for K2-32B and K2-65B (Appendix B). Across the five checkpoints, prediction MAE remains within 0.67–1.19 pp (Figure 3, left).

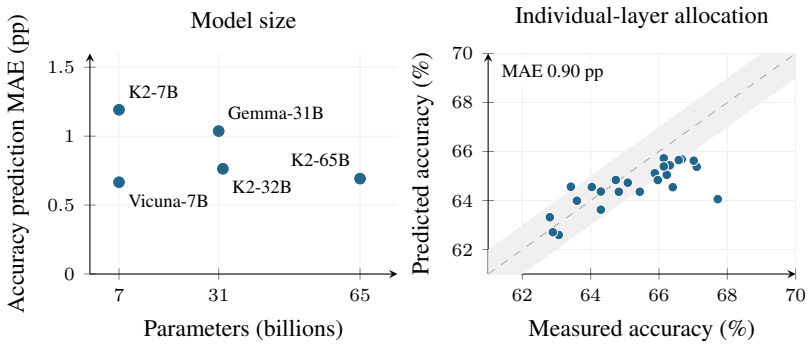

Figure 3: GPTQ accuracy prediction on MMLU. Left: MAE versus model size at nominal 3.5 bits (complete spaces for the four 7B–32B checkpoints; 16 of 20 allocations for K2-65B). Right: predicted versus measured accuracy for 24 presampled K2-7B individual-layer allocations; dashed line: equality; gray band: ± 1 pp.

4.7 Composing compression at individual-layer granularity

To test whether the model composes finer-grained compression effects, we use individual layers as blocks. K2-Horizon-7B's 36 decoder layers give 72 W3/W4 parameter pairs; assigning W4 to 18 layers creates $\binom{36}{18} = 9{,}075{,}135{,}300$ feasible allocations. Each prediction combines 36 selected pairs from this cache without candidate-specific model execution. We evaluate 24 allocations sampled uniformly before measuring their compressed responses, assembling each from cached layer options. Their measured accuracies span 4.91 pp. At individual-layer granularity, the direct-measurement predictor achieves 0.90 pp prediction MAE and $\rho = 0.686$ (Figure 3, right).

4.8 Turning predictions into compression decisions

We finally examine whether accuracy predictions guide compression decisions under precision and sparsity budgets, for single-task and multitask objectives.

Placing precision where it preserves task accuracy. To test whether predictions identify the best precision allocation, we maximize predicted MMLU selection accuracy over the complete Gemma-31B GPTQ space. The selected allocation assigns W4 to interior blocks 2–4; assigning W4 to the earliest three blocks gives 79.74% test accuracy. The selected allocation is best in the space at 82.46% test accuracy, a 2.72 pp improvement (95% paired bootstrap confidence interval: 1.14–4.39 pp).

Allocating sparsity under a common pruning procedure. We next test whether maximizing predicted task accuracy selects better sparsity allocations than OWL (Yin et al., 2024). OWL distributes sparsity using layerwise outlier statistics, while our predictor evaluates candidate allocations by

composing local margin statistics. Using single-block statistics measured with dense-model activations, we maximize predicted MMLU selection accuracy over block sparsities in $\{0.3, 0.5, 0.6, 0.7\}$ under the target weight budget. We then apply the selected assignments unchanged with sequential Wanda, which propagates activations through previously pruned layers. Both methods use this pruning procedure with the same calibration texts and targeted weights, under matched sparsity budgets within each model. OWL uses threshold multiplier $M = 5$, with $\lambda = 0.08$ controlling the spread of layerwise sparsities.

Table 4 shows MMLU gains over OWL of 3.33 pp on K2-Horizon-7B at 50% sparsity and 26.84 pp on Gemma-4-12B PT at 60% sparsity. The same MMLU-selected allocations also outperform OWL on BoolQ and OpenBookQA for both models, without task-specific reselection.

Table 4: Test accuracy (%) after sequential Wanda pruning. Realized sparsities differ by less than 0.04 pp within each model.

Model	Sparsity	Task	OWL	Ours
K2-Horizon-7B	50%	MMLU	61.93	65.26
		BoolQ	83.68	84.78
		OpenBookQA	40.80	41.20
Gemma-4-12B PT	60%	MMLU	32.72	59.56
		BoolQ	58.07	66.99
		OpenBookQA	26.00	34.40

Selecting allocations for multiple tasks. To test whether predictions guide multitask allocation, we retrospectively evaluate all 12 Vicuna-7B GPTQ allocations at 3.50 bits using an equal-weight objective over ARC-Challenge, HellaSwag, PIQA, and BoolQ. Direct block measurements supply task-specific margin statistics. Predictions of the four-task test mean have 0.74 pp MAE across the space. Maximizing the corresponding predicted mean on the selection splits selects an allocation within 0.02 pp of the highest measured four-task test mean.

5 Conclusion

Our model predicts complete-model accuracy by composing local measurements of margin scaling and residual variation, then evaluating answer competition on a labeled reference set scored by the dense model. Complete-space experiments show that these statistics track accuracy across model families and compression methods. Across five K2-Horizon-7B quantization spaces, composing margin statistics reduces mean per-space prediction MAE relative to both scalar baselines. The fitted Gaussian predictor also has lower prediction MAE than both fitted residual alternatives in each of six quantization spaces. The same construction extends across decoder models from 7B to 65B, individual-layer combinations, multiple-choice tasks, and encoder-based classification. Maximizing predicted accuracy identifies the best measured allocation in the complete Gemma-31B quantization space; the MMLU-selected sparsity allocations outperform OWL on MMLU, BoolQ, and OpenBookQA under matched budgets. The same accuracy objective supports multitask allocation, as illustrated by our retrospective Vicuna-7B analysis.

Reproducibility statement

Code is available at https://anonymous.4open.science/r/predict-before-compress-1D01/. The appendices specify parameter estimation, compression settings, calibration data, and the evaluation protocol.

AI use statement

Generative AI assisted with conceptual presentation, interpretation and auditing of experimental records, manuscript drafting and editing, bibliographic verification, and preparation of analysis scripts, tables, and figures. The reported compression experiments were conducted separately from manuscript preparation.

REFERENCES

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Convai Innovations. Laya: Model card. https://huggingface.co/convaiinnovations/laya, 2026. English checkpoint, accessed September 24, 2026.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael Mahoney, and Kurt Keutzer. HAWQ: Hessian AWare quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. URL https://arxiv.org/abs/1905.03696.

Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ-V2: Hessian aware trace-weighted quantization of neural networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/d77c703536718b95308130ff2e5cf9ee-Abstract.html.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10323–10337. PMLR, 2023. URL https://proceedings.mlr.press/v202/frantar23a.html.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.17323.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, July 2024. URL https://zenodo.org/records/12608602.

Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026. URL https://arxiv.org/abs/2607.02770.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. SliM-LLM: Salience-driven mixed-precision quantization for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25672–25692. PMLR, 2025. URL https://proceedings.mlr.press/v267/huang25aa.html.

IFM Team. Introducing K2 Horizon: Frontier performance, radically open, 2026. URL https://ifm.ai/blog/k2/.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, 2024. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/42a452cbafa9dd64e9ba4aa95cc1ef21-Abstract-Conference.html.

Zhengzhong Liu, Bowen Tan, Hongyi Wang, Willie Neiswanger, Tianhua Tao, Haonan Li, Fajri Koto, Yuqi Wang, Suqi Sun, Omkar Pangarkar, Richard Fan, Yi Gu, Victor Miller, Liqun Ma, Liping Tang, Nikhil Ranjan, Yonghao Zhuang, Guowei He, Renxi Wang, Mingkai Deng, Robin Algayres, Yuanzhi Li, Zhiqiang Shen, Preslav Nakov, and Eric Xing. LLM360 K2: Building a 65b 360-open-source large language model from scratch, 2025. URL https://arxiv.org/abs/2501.07124.

Vladimir Malinovskii, Andrei Panferov, Ivan Ilin, Han Guo, Peter Richtárik, and Dan Alistarh. HIGGS: Pushing the limits of large language model quantization via the linearity theorem. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 10857–10886. Association for Computational Linguistics, April 2025. doi: 10.18653/v1/2025.naacl-long.543. URL https://aclanthology.org/2025.naacl-long.543/.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399. URL https://ojs.aaai.org/index.php/AAAI/article/view/6399.

Oliver Sieberling, Denis Kuznedelev, Eldar Kurtic, and Dan Alistarh. EvoPress: Accurate dynamic model compression via evolutionary search. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55556–55590. PMLR, 2025. URL https://proceedings.mlr.press/v267/sieberling25a.html.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.11695.

David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005. URL https://stanford.edu/~dntse/wireless_book.html.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2526–2547, 2025. URL https://aclanthology.org/2025.acl-long.127/.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Kumar Jaiswal, Mykola Pechenizkiy, Yi Liang, Michael Bendersky, Zhangyang Wang, and Shiwei Liu. Outlier weighed layerwise sparsity (OWL): A missing secret sauce for pruning LLMs to high sparsity. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57101–57115. PMLR, 2024. URL https://proceedings.mlr.press/v235/yin24e.html.

Ben Zandonati, Adrian Alan Pol, Maurizio Pierini, Olya Sirkin, and Tal Kopetz. FIT: A metric for model sensitivity. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PDG4-Y3aboN.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, pp. 649–657, 2015. URL https://papers.nips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

A Derivations

Combining blockwise errors. For $v_b = \alpha_b v_{b-1} + \eta_b$, repeated substitution gives

$$v_L = \left(\prod_{b=1}^{L} \alpha_b \right) v_0 + \sum_{b=1}^{L} \eta_b \prod_{j>b} \alpha_j. \tag{7}$$

Writing $q_b = \prod_{j>b} \alpha_j$, the effective variance is

$$\operatorname{Var}\left(\sum_b q_b \eta_b \right) = \sum_b q_b^2 \operatorname{Var}(\eta_b) + 2 \sum_{b<j} q_b q_j \operatorname{Cov}(\eta_b, \eta_j). \tag{8}$$

The uncorrelated-error model gives Equation 5. The complete-space experiments test this composition model using statistics measured with each block compressed separately.

B Evaluation and implementation details

Checkpoints and scoring. The checkpoint identifiers are `IFM/K2-Horizon-7B`, `IFM/K2-Horizon-32B`, `lmsys/vicuna-7b-v1.5`, `google/gemma-4-12B`, `google/gemma-4-31B`, and `IFM/K2`. Answer scores are continuation log-likelihoods; byte normalization divides by the continuation's UTF-8 byte length. PIQA, HellaSwag, OpenBookQA, and ARC-Challenge use byte-normalized scores; BoolQ, WinoGrande, and MMLU use unnormalized scores. HellaSwag is capped at 2,000 items. Runs on the Gemma checkpoints preserve each model's beginning-of-sequence handling.

Block partitions and feasible spaces. K2-Horizon-7B uses six blocks of six layers. Its GPTQ spaces assign W4 to two, three, or four blocks, yielding 15, 20, or 15 allocations at 3.33, 3.50, or 3.67 bits per quantized weight. AWQ and rotated GPTQ use the 20-allocation space. Vicuna-7B uses blocks of 6/5/5/5/5/6 layers; its 12 feasible allocations assign W4 to exactly 16 of 32 layers. Gemma-12B uses eight six-layer blocks; assigning 70% sparsity to two and 50% to six gives 28 allocations at 55% sparsity. K2-7B's 60%-sparsity space contains all 20 assignments with three blocks at 50% and three at 70%, together with uniform 60% pruning. Metadata and unchanged components are excluded from targeted-weight storage accounting.

For larger models, the six block lengths are 11/10/11/11/10/11 (K2-Horizon-32B), 10/10/10/10/10/10 (Gemma-31B), and 13/14/13/13/14/13 (K2-65B). Assigning three blocks

to W4 gives layer-weighted average bit widths of 3.4844-3.5156, 3.5000, and 3.4875-3.5125 bits, respectively. We refer to these as nominal 3.5-bit spaces; the first and third have small differences in average bit width due to unequal block lengths. Each model uses its own 12 direct GPTQ block measurements.

Task suite. The task-specific measurements cover BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), ARC-Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2020), and MMLU. Tasks other than MMLU use zero-shot prompts. The non-MMLU complete-space predictions use direct measurements fitted separately for each task. The AG News encoder experiment uses the encoder classification protocol in Appendix E.

Partitions. The selected evaluation items are partitioned 25/25/50 into fit, selection, and test splits, stratified by task and, for MMLU, subject. Each MMLU subject contributes 10 fit, 10 selection, and 20 test items. Stable keys use task, subtask, and document identifier; seed 20260919 fixes their assignment. We fit block statistics on the fit split and select configurations on the selection split. Reference labels enter Equation 2 alongside dense-model scores; compressed test responses supply the reported accuracy measurements.

Error model estimation and numerical references. The implementation adds a small positive constant to the denominator in Equation 3 and computes the empirical residual standard deviation with denominator n in Equation 4. Predictions in the K2-Horizon-7B GPTQ 3.50-bit space use 128 draws per item; the adjacent 3.33- and 3.67-bit budgets use 256. Seed 1 supplies common Gaussian draws within corresponding prediction calls.

Pruning comparison. K2-7B uses six blocks; Gemma-12B PT uses eight blocks of six layers. For selection, the predictor uses single-block Wanda statistics with activation statistics collected from the uncompressed model. The selector enumerates block sparsity assignments on the grid $\{0.3, 0.5, 0.6, 0.7\}$ and maximizes predicted MMLU selection accuracy. The selected block sparsities are $[0.3, 0.3, 0.5, 0.6, 0.7, 0.6]$ for K2-7B and $[0.7, 0.7, 0.5, 0.5, 0.3, 0.7, 0.7, 0.7]$ for Gemma-12B PT.

For the comparison with OWL in Table 4, we transfer these configurations unchanged to sequential Wanda with updated upstream activations. K2-7B uses 103 eligible calibration texts and Gemma-12B PT uses 120, identical within each comparison. The measured sparsities are 0.499899/0.499899 for K2-7B and 0.599996/0.599629 for Gemma-12B PT (ours/OWL). The OWL implementation counts activation-weighted weight scores above $M = 5$ times the layer mean. It rescales the resulting layerwise outlier ratios to a range of width 2λ, with $\lambda = 0.08$, then shifts their mean to the target retention rate, one minus the target sparsity. Its layer allocation and the accuracy-based allocation across blocks operate on the same targeted weight tensors.

C Scalar accuracy-composition baselines

Additive accuracy prediction. Let $A^{\text{select}}_{b,a}$ be the accuracy measured with only block b compressed using option a, and $A^{\text{select}}_{b,0}$ the dense model's accuracy on the same selection items. With $\delta_{b,a} = A^{\text{select}}_{b,0} - A^{\text{select}}_{b,a}$, the additive prediction is

$$\widehat{A}^{\text{test}}_{\text{add}}(\boldsymbol{a}) = A^{\text{test}}_{\text{dense}} - \sum_{b} \delta_{b,a_b}. \tag{9}$$

This additive approximation has a fixed intercept equal to the uncompressed accuracy: it retains each measured single-block effect and sets all cross-block interaction terms to zero. In every space, we use the corresponding dense model's test accuracy as the intercept and recompute the comparison from the saved direct measurements. The additive selector minimizes $\sum_b \delta_{b,a_b}$.

Multiplicative accuracy prediction. The second scalar baseline multiplies the ratios of single-block to uncompressed-model accuracy:

$$\widehat{A}^{\text{test}}_{\text{ret}}(\boldsymbol{a}) = A^{\text{test}}_{\text{dense}} \prod_b \frac{A^{\text{select}}_{b,a_b}}{A^{\text{select}}_{b,0}}. \tag{10}$$

It uses exactly the same single-block accuracies and dense-model accuracy on the test reference set as Equation 9, but assumes separability in log accuracy rather than accuracy. Both baselines compose task-level scalars, while the predictor composes an answer-score model.

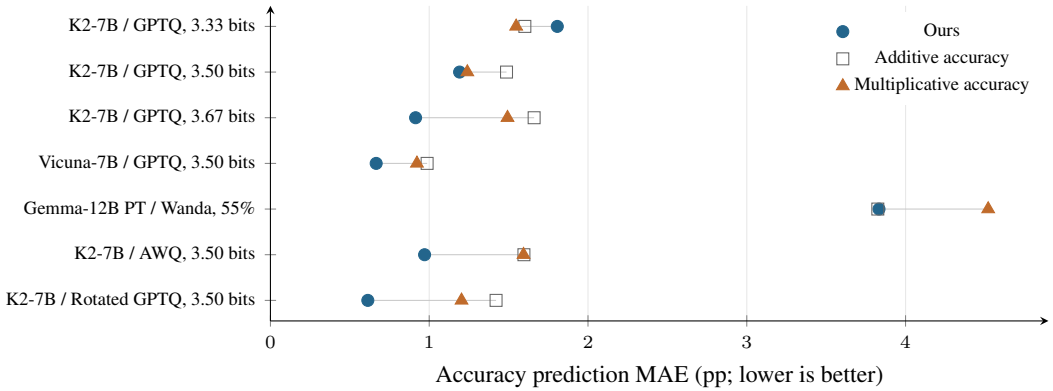

Figure 4: MMLU accuracy-prediction MAE across seven complete allocation spaces, using direct measurements for all predictors.

D Residual and noise-model diagnostics

For each single-block measurement, we fit α on the fit split and form the regression residual r_i for the dense top-two margin in Equation 4. The 448 measurements of block options across tasks come from the K2-7B GPTQ, AWQ, and rotated-GPTQ experiments, Vicuna-7B GPTQ, and Gemma-12B PT Wanda, each evaluated on seven tasks. Across these measurement sets, the median $|\bar{r}|/\sigma$ is 0.028 (95th percentile 0.094), and 16 of 448 bootstrap 95% confidence intervals for $\bar{r}$ exclude zero. The median absolute skewness is 0.277 and median excess kurtosis is 2.32. For the 64 MMLU measurement sets, the corresponding shape statistics are 0.22 and 1.44. The Gaussian predictor uses the fitted scaling coefficient and residual standard deviation to approximate decision probabilities.

To examine dependence among answers, we form a score-residual vector by subtracting the fitted scaling coefficient times the dense score vector from the compressed score vector. We then subtract each item's mean residual across its four answer choices. Across 24 K2-7B AWQ and rotated-GPTQ MMLU measurement sets, the median within-set off-diagonal Pearson correlation is -0.342 (10th-90th percentile of measurement-set medians: -0.354 to -0.321). This is near the $-1/3$ correlation induced algebraically when four exchangeable score residuals are centered to sum to zero. The median max/min ratio of variances of pairwise margin residuals is 1.30 for AWQ and 1.37 for rotated GPTQ; across all 64 MMLU measurement sets it is 1.37.

We next assess how these residual characteristics affect accuracy prediction by comparing three noise models fitted to block measurements and a mean-shift variant (Table 5). The Gaussian model with estimated covariance uses a shrinkage estimate of each single-block fit-split covariance of answer-score residuals, then composes those matrices with later scaling coefficients. The empirical variant composes resampled fit-split residual vectors. The covariance variant retains the empirical answer-score residual scale, while the independent variant estimates its scale from top-two margin residuals. The comparison therefore assesses the fitted predictors as a whole on fully compressed test configurations. The mean-shift variant adds the composed fit-split residual mean to the dense model's scaled top-choice score before simulation. All variants retain the same combined scaling coefficients.

The diagnostic covers six complete quantization spaces and 102 allocations, with direct block measurements and 96 Monte Carlo draws per item for every variant. This matched rerun is reported

separately from Table 2. We evaluate accuracy-prediction MAE and two item-level probability metrics: expected calibration error (ECE) and binary Brier score. Both probability metrics compare the predicted probability that a compressed model answers an item correctly with the observed correctness indicator. Within each space, we pool all item–configuration pairs; ECE uses 10 equal-width probability bins. Probabilities are clipped to $[10^{-4}, 1 - 10^{-4}]$ before these metrics are evaluated.

The independent Gaussian predictor has lower MMLU accuracy-prediction MAE than the covariance-based Gaussian and empirical-residual predictors in all six quantization spaces. Averaged across spaces, its item-level ECE is 0.0189, compared with 0.0216 for the Gaussian model with estimated covariance and 0.0238 for the empirical bootstrap. Binary Brier scores are 0.1088, 0.1087, and 0.1092, respectively.

Table 5: MMLU accuracy-prediction MAE (pp) in a matched diagnostic rerun with shared scaling coefficients and 96 draws per item. IID denotes independent and identically distributed Gaussian score residuals. N: number of allocations. The final row reports unweighted mean MAEs across spaces and the total allocation count.

Space	N	IID	Covariance	Empirical	IID + mean shift
K2-7B GPTQ, 3.33 bits	15	1.77	2.28	2.24	1.78
K2-7B GPTQ, 3.50 bits	20	1.23	1.51	1.49	1.16
K2-7B GPTQ, 3.67 bits	15	0.90	1.22	1.30	0.91
Vicuna-7B GPTQ, 3.50 bits	12	0.66	1.14	1.11	0.81
K2-7B AWQ, 3.50 bits	20	1.01	1.39	1.33	0.95
K2-7B rotated GPTQ, 3.50 bits	20	0.60	0.77	0.81	0.60
Mean across spaces	102	1.03	1.38	1.38	1.03

E EXPERIMENTAL DEVELOPMENT AND EVALUATION PROTOCOL

The complete-space experiments use the fixed fit/selection/test item partition described in Section 4. The roles of the spaces are as follows:

- K2-7B GPTQ, 3.50 bits: a complete 20-allocation space for evaluating predictions from reusable block measurements.
- K2-7B GPTQ, 3.33 and 3.67 bits: evaluation at adjacent budgets. The 30 evaluations include 27 new allocations and repeats of `443333`, `333344`, and `333434`.
- Vicuna-7B GPTQ, 3.50 bits: evaluation across all 12 allocations satisfying the unequal-block budget, using direct block measurements from the earlier model comparison.
- Gemma-12B PT Wanda, 55% sparsity: evaluation of the composition rule on 28 new allocations, with predictions saved before complete-configuration evaluation.
- K2-7B AWQ and rotated GPTQ, 3.50 bits: evaluation of the composition rule after collecting 12 block measurements for each compression method. AWQ includes one allocation (`444333`) previously run on a 285-item pilot, of which 148 items overlap the final test reference, and 19 newly constructed allocations. Excluding the AWQ pilot allocation gives approximately 1.00 pp MAE and $\rho = 0.910$. Rotated GPTQ contains 20 new allocations.

These seven spaces comprise 130 complete-configuration evaluations across precision budgets, model families, and compression methods.

The K2-7B Wanda space comprises 18 newly evaluated allocations and three reused configurations, for 21 total. Its predictions use the previously measured block statistics.

Final model-scale and layerwise experiments. The larger-model manifests save predictions based on direct measurements before complete-configuration evaluation. K2-Horizon-32B and Gemma-31B include all 20 three-W4 assignments; K2-65B includes the first 16 evaluated assignments in the prescribed run order.

The individual-layer experiment caches separate GPTQ W3 and W4 options for all 36 K2-7B layers, using calibration inputs from the uncompressed model to construct each option. Candidates are

assembled from these cached options. A random seed of 2026 fixes 24 uniformly sampled allocations with exactly 18 W4 layers before test evaluation. Each candidate's prediction combines its 36 directly estimated parameter pairs and applies the accuracy mapping using dense-model scores and correct-answer labels from the test reference set. Figure 5 shows the configuration-level predictions for the larger models and individual-layer allocations.

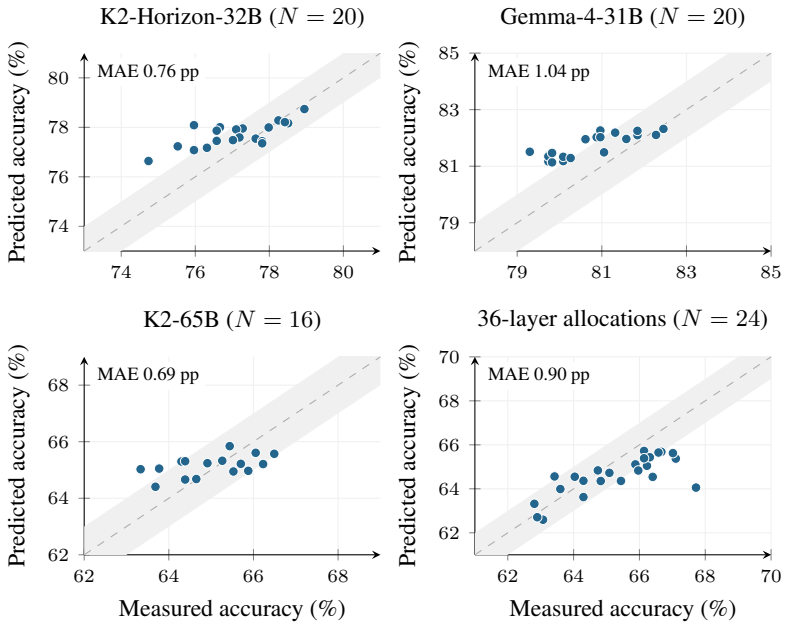

Figure 5: Nominal 3.5-bit GPTQ scaling predictions on MMLU. K2-32B and Gemma-31B spaces are complete; K2-65B shows 16 evaluated allocations; the layerwise panel shows 24 presampled K2-Horizon-7B allocations. Dashed lines: equality; gray bands: ± 1 pp.

Non-MMLU prediction and joint selection. For the K2-7B GPTQ 3.50-bit space, task-specific test predictions for BoolQ, HellaSwag, and WinoGrande were saved before the complete configurations were built. These predictions use parameters fitted from direct single-block measurements on each task's fit split. Separately, a retrospective application of the fixed selector to the selection reference identifies the best measured BoolQ allocation.

The retrospective four-task Vicuna-7B analysis uses all 12 GPTQ allocations at 3.50 bits. Direct block measurements on each task's fit split supply its margin statistics. We use 64 Gaussian draws per item to predict selection accuracy and 128 to predict test accuracy, averaging equally over ARC-Challenge, HellaSwag, PIQA, and BoolQ. The four-task and MMLU-only objectives both select `444333`, which ranks second by measured four-task test mean, 0.014 pp below the best allocation, `344334`.

Encoder classification protocol. We use the English checkpoint `convaiinnovations/laya`. The AG News experiment uses its four class logits for World, Sports, Business, and Sci/Tech and measures accuracy by comparing their argmax with the correct class label. Seed 2026 selects 2,000 evaluation items; the fixed 25/25/50 split gives 500 fit, 500 selection, and 1,000 test items. Compression targets linear weights in the 28 encoder layers. Cached GPTQ W3/W4 and Wanda 50%/70% options use statistics collected from the uncompressed encoder on 128 training sequences, with 32 each from AG News, emotion classification, SST-5, and BoolQ. GPTQ uses groups of 128 weights; Wanda prunes within each output row. Embeddings, normalization parameters, and the decision head remain uncompressed.

Each four-layer block has two options per compression method. Choosing three or four of seven blocks at W4 gives 35 allocations at each of 3.43 and 3.57 bits. Choosing three or four blocks at

70% sparsity and the remainder at 50% gives 35 allocations at each nominal sparsity of 58.57% and 61.43%. Parameters are fitted from direct score measurements on the 500 fit items.

For individual-layer prediction, 56 measurements supply W3/W4 parameters for all 28 layers, and seed 2026 fixes 100 random allocations with 14 W4 layers. The four complete block spaces and the layerwise sample together yield 240 AG News predictions. All were saved before their configurations were evaluated and use the composition rule with 200 Gaussian draws. The four previously evaluated uniform references and the two selected layerwise configurations are separate from these complete spaces and random-sample results.

Additional compression settings. Our AWQ implementation uses the llm-compressor scale search with grid size 20, followed by asymmetric round-to-nearest quantization with group size 128. The rotated implementation uses randomized Hadamard transforms on the input and output dimensions of the targeted linear layers followed by symmetric GPTQ with group size 128. Each implementation supplies its own cached W3/W4 measurements on the same fit items. Reported predictions therefore reuse the composition rule with compression-method-specific parameters.

F Measurement cost and reuse

Table 6: Vicuna-7B GPTQ timing (GPU-minutes, A100-80GB), averaged over three runs; loading excluded. Block measurements construct 12 variants and score each on 2,280 MMLU reference items. Candidate costs cover one complete allocation evaluated on the same number of items.

Procedure	Construction	Measurement / evaluation	Total
Block measurements	34.93	36.53	71.47
One complete candidate	16.77	2.97	19.74

For Table 6, we difference cumulative timestamps from runs 4-6 to separate construction from measurement. GPU-minutes sum the elapsed durations of the single-GPU runs. Only the fit partition is used for parameter estimation. Loading the model for direct measurement takes an additional 3.03 minutes.

For K2-7B at individual-layer granularity, constructing the 72-option GPTQ cache takes 28.39 GPU-minutes, and measuring its margin statistics takes 85.61 GPU-minutes. Thereafter, composing a candidate costs $O(L)$ scalar operations; its predicted accuracy uses dense-model reference scores, correct-answer labels, and Gaussian draws, with no compressed-model execution.

G Additional precision budgets

The same directly measured block statistics predict all allocations in the K2-7B GPTQ spaces at 3.33, 3.50, and 3.67 bits, with accuracy-prediction MAEs of 1.81, 1.19, and 0.91 pp, respectively.

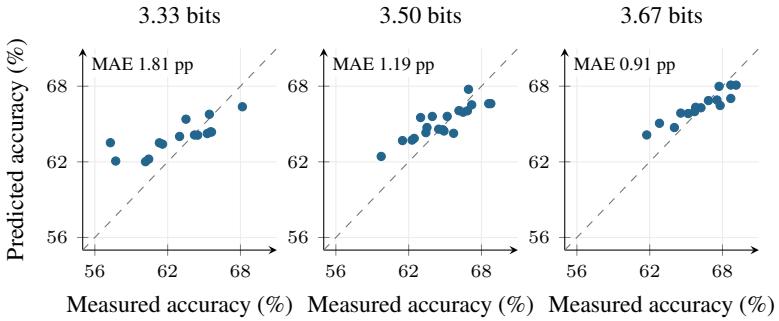

Figure 6: Complete K2-Horizon-7B GPTQ allocation spaces on MMLU: 15, 20, and 15 configurations, from left to right. Dashed lines denote equality.

H ADDITIONAL COMPLETE ALLOCATION SPACES

The Gemma-4-12B PT Wanda experiment uses eight six-layer blocks; assigning 70% sparsity to two and 50% to six gives all 28 allocations at 55% sparsity. Their measured MMLU accuracies span 36.1 pp. Composing the direct block measurements gives 3.83 pp prediction MAE and $\rho = 0.956$. Figure 7 also shows the complete 20-allocation BoolQ experiment described in Section 4.5, which gives 0.71 pp MAE using direct measurements and the composition rule.

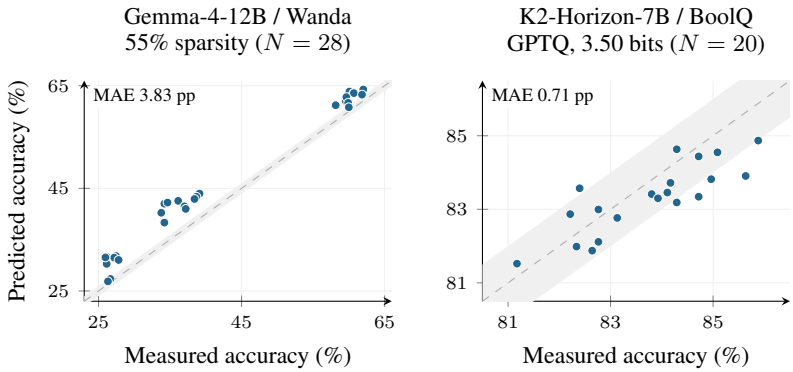

Figure 7: Complete-space accuracy predictions for Gemma-12B PT/MMLU and K2-7B/BoolQ. Dashed lines: equality; gray bands: ± 1 pp.