How Much do Language Models Regurgitate under Post-Training Quantization?

Anonymous authors
Paper under double-blind review

Abstract

Quantization has become a standard practice for efficient deployment of today's large language models. However, there is still a lack of understanding of its impact on verbatim regurgitation—where models may regurgitate training data samples, which can be a privacy vulnerability. In this work, we present the first comprehensive study of how post-training quantization affects verbatim regurgitation in LLMs, uncovering multiple key insights. First, we show that reducing precision consistently lowers regurgitation rate across all model sizes, but at the cost of decreased downstream performance. Second, we find that for the same set of sequences, larger quantized models regurgitate more than smaller models, consistent across varying prefix & suffix lengths. We also propose a scaling law to model verbatim regurgitation under quantization. Lastly, fine-grained layer-wise and component-wise analyses reveal a consistent trend: quantizing attention modules leads to a higher decrease in regurgitation compared to feed-forward networks, and quantizing later layers has a more pronounced impact than early-middle layers, further supported by flip rate analyses. Our empirical study covers three modern post-training quantization methods and six models from the Pythia suite, providing novel insights into the impact of quantization on verbatim regurgitation.

1 Introduction

Large Language Models (LLMs) are trained on diverse data sources collected from the internet and exhibit strong capabilities in language modeling, understanding, and reasoning (Bubeck et al., 2023; Liu et al., 2023). However, these data sources may contain privacy-sensitive information, such as personal contact details, social security numbers, or date of birth (Lukas et al., 2023; Dodge et al., 2021). During training, LLMs can undesirably store such private information and later regurgitate training samples at inference time, leading to a potential leakage of sensitive information (Nasr et al., 2023; Staab et al., 2023).

On the other hand, the impressive performance of these models comes with a substantial increase in model size (Kaplan et al., 2020). This rapid growth significantly limits their deployment on consumer devices and in resource-constrained environments due to high memory requirements and energy consumption (Alizadeh et al., 2024). To address this issue, quantization has been widely adopted for compressing LLMs (Nagel et al., 2021; Gholami et al., 2022). Quantization reduces the numerical precision of model parameters and/or activations (e.g., from 32-bit floating point to 8-bit or 4-bit representations), reducing the memory footprint and improving computational efficiency (Bondarenko et al., 2023; Kuzmin et al., 2023), accompanied by minimal degradation in downstream task performance. In many cases, significantly smaller quantized models (e.g., reducing the effective footprint of a 10B-parameter model to that of a much smaller model like 4B) can achieve performance comparable to their full-precision counterparts on standard benchmarks (Frantar et al., 2022; Badri & Shaji, 2023). In addition to efficiency gains, recent studies have attempted to study the impact of quantization on different trust dimensions such as stereotype, toxicity, fairness, and robustness (Hong et al., 2024; Bellam & Kim, 2025; Liu et al., 2025; Chen et al., 2023). However, its effect on verbatim regurgitation of training data remains unexplored. We tackle this gap, as language models can potentially regurgitate training data, including privacy-sensitive content, and we genuinely wonder whether quantization-induced compression may influence this behavior or not.

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.

Therefore, through this paper we aim to understand the relationship between quantization (post-training quantization in this paper's scope) and verbatim regurgitation in language models, ***which to the best of our knowledge, has never been explored in literature***. Please note that the aim of this paper is to provide novel insights on the impact of quantization on verbatim regurgitation to broaden the understanding, but not to provide a particular solution. In this paper, we conduct analysis across three modern post-training quantization methods–HQQ, BitsAndBytes, and GPTQ, and six models in the Pythia model suite, spanning sizes of 70M, 160M, 410M, 1.4B, 2.8B and 6.9B. We summarize the key insights and contributions of our paper as follows:

- **Verbatim regurgitation reduces with decrease in quantization precision:** We show that as we decrease the quantization precision level, verbatim regurgitation also decreases.
- **Larger models regurgitate more than smaller models after quantization.** For the same set of sequences, larger models regurgitate more sequences than smaller models after quantization.
- **The proposed scaling law captures how verbatim regurgitation behaves under quantization.** We propose a scaling law for verbatim regurgitation under quantization based on precision, model size and prefix-length.
- **Quantizing either Attention modules or Later layers impacts regurgitation more.** We observe a consistent trend where quantizing the attention module (not feed-forward) or later layers (neither early nor middle) has a higher influence in reducing regurgitation, with all the trends well supported by the Flip Rate metric analyses.

2 Related Work

Model Quantization: In the current era of large language models (LLMs), quantization has become a standard and default approach for compressing them, enabling reduced memory footprint and improved inference efficiency (Nagel et al., 2021; Gholami et al., 2022). Broadly, existing methods fall into two categories: Post-Training Quantization (PTQ) (Yao et al., 2022; Zhao et al., 2025), which applies quantization just after model training, and Quantization-Aware Training (QAT) (Chen et al., 2025; Liu et al., 2024), which incorporates quantization effects during training. Building on these paradigms, a range of techniques—such as GPTQ (Frantar et al., 2022), AWQ (Lin et al., 2024), HQQ (Badri & Shaji, 2023), BitsAndBytes (Dettmers et al., 2022), BRECQ (Li et al., 2021), and SmoothQuant (Xiao et al., 2023)—have been proposed to achieve increased compression while preserving downstream task performance. Beyond efficiency, recent works have tried to examine the implications of quantization on different trust dimensions including model toxicity, fairness, robustness, catastrophic forgetting, and machine unlearning (Hong et al., 2024; Bellam & Kim, 2025; Liu et al., 2025; Chen et al., 2023; Zhang et al., 2024).

Verbatim Regurgitation: LLMs are trained on heterogeneous corpora that may include sensitive or personally identifiable information (PII), such as phone number, social security number, or date of birth (Dodge et al., 2021; Lukas et al., 2023). During training, such models can unintentionally store such data points in their weights and reproduce them verbatim at inference time (Brown et al., 2022; Carlini et al., 2021). This phenomenon, referred to as verbatim regurgitation, raises concerns around unintended privacy leakage. Recent work has focused on formalizing regurgitation (Carlini et al., 2022), how it varies across models (Nasr et al., 2023), and identifying regions and mechanisms responsible for it within the network (Huang et al., 2024; Chang et al., 2024). In parallel, a growing body of research has examined how model efficiency interventions—such as pruning (Ni et al., 2025), knowledge distillation (Borkar et al., 2026), and parameter-efficient fine-tuning (e.g., LoRA) (Wang & Li, 2025; Bossy et al., 2025)—influence the verbatim regurgitation behavior. However, no prior work has investigated the implication of quantization (post-training quantization) on verbatim regurgitation, which remains equally important due to the potential impact of regurgitation on revealing privacy-sensitive data during inference.

3 Preliminaries

3.1 Background on Quantization

Quantization is a compression technique that maps high-precision real-valued parameters to a finite set of discrete levels, enabling efficient storage and computation (Nagel et al., 2021; Bondarenko

et al., 2023). In modern deep learning systems, quantization is typically parameterized by scale, zero-point, and target bit-width, to define the mapping between real and low-precision values.

Formally, let $\mathbf{x} \in \mathbb{R}^d$ denote a real-valued vector. Given a scale $s > 0$, zero-point $z \in \mathbb{Z}$, and bit-width b, we define the quantization & de-quantization operations as:

$$\mathbf{x}_{\mathrm{q}} = \mathrm{clamp}\left(\left\lfloor \frac{\mathbf{x}}{s} \right\rceil + z,\ 0,\ 2^b - 1\right) \qquad \& \qquad \hat{\mathbf{x}} = s\left(\mathbf{x}_{\mathrm{q}} - z\right), \tag{1}$$

where $\lfloor \cdot \rceil$ denotes round-to-nearest operator, $\mathrm{clamp}(\cdot)$ restricts values to a specified range, and $\hat{\mathbf{x}}$ is the real-valued approximation of the original vector $\mathbf{x}$.

3.2 Type of Quantization Technique Used: Post-Training Quantization

In this paper, we specifically focus on Post-Training Quantization (PTQ) techniques. We do not consider Quantization-Aware Training (QAT), as it requires additional retraining, in contrast to PTQ, which operates directly on pre-trained models, making it a more practical and widely adopted approach for compressing large-scale models in practice (Gholami et al., 2022). The three PTQ techniques considered in this paper are: BitsAndBytes (Dettmers et al., 2022), GPTQ (Frantar et al., 2022), and HQQ (Badri & Shaji, 2023). In this paper, we consider different precision levels ranging across FP32, FP16, INT8, INT4, INT3, and INT2. Also, please note that throughout this work, whenever we use the term quantization, we are specifically referring to PTQ.

3.3 Verbatim Regurgitation & Metrics

As discussed in prior literature (Carlini et al., 2022; Nasr et al., 2023), verbatim regurgitation refers to the phenomenon where a model reproduces exact training set samples when prompted with their corresponding prefixes. Formally, it is defined as follows (Carlini et al., 2022):

Definition. A training sample s is said to be regurgitated (or extracted) if, when the model is prompted with prefix string p of length k, it generates a suffix y such that the concatenation $[p \,\|\, y]$ is identical to the training sample s, under greedy decoding.

To study the impact of quantization on verbatim regurgitation, we define the following two metrics:

- **Verbatim Regurgitation (%)** Computes the proportion of samples that are extracted under the Verbatim Regurgitation definition.
- **Test Accuracy (%)** Measures the performance of the language models under downstream tasks.

3.4 Output Logits Analysis via Flip Rate

To further understand the impact of quantization on regurgitation, we analyze how quantization affects the output logits for a given input prefix p corresponding to a regurgitated sequence. This is because changes in the output logits can potentially disrupt verbatim reproduction of training sequences by altering token-level predictions. Prior work has tried to study the impact of quantization on output logits using metrics such as Signal-to-Quantization-Noise Ratio (SQNR) (Kuzmin et al., 2023; Liu et al., 2024) and KL-divergence (Kong et al., 2026; Dutta et al., 2024). These approaches typically operate under a teacher-forcing setting (Dutta et al., 2024; Boizard et al., 2024), where both the full precision (FP32) and quantized models are conditioned on the same ground-truth token sequence, enabling token-wise comparison of predicted probability distributions at each position across the quantized and FP32 models. However, these metrics primarily capture distributional shifts in the overall probability distributions rather than their direct impact on decoding behavior which is essential to study whether a model regurgitates the training samples. Hence, they are not effective in explaining the impact of quantization on verbatim regurgitation–more details are discussed in Appendix A.

Alternatively, we propose an adequate, behaviorally grounded metric, termed as **FLIP RATE**, which directly measures the extent to which quantization disrupts verbatim regurgitation of a training sample under a teacher-forcing evaluation setting. This ensures a fair comparison between the original FP32 model θ and a quantized model $\hat{\theta}$ by conditioning both on the same ground-truth sequence. Formally Flip Rate *(FR(S))* for a sequence S is defined as:

$$FR(S) = \frac{1}{T} \sum_{t=1}^{T} I_t(S) \tag{2}$$

where, t is the suffix token position, T is the total number of suffix tokens, $I_t(S) = \mathbb{I}\left[\arg\max_i z_{i,t}(\theta, S) \neq \arg\max_i z_{i,t}(\hat{\theta}, S)\right]$ such that $I_t(S) = 1$ if $\arg\max_i z_{i,t}(\theta, S) \neq \arg\max_i z_{i,t}(\hat{\theta}, S)$ and 0 otherwise, i denotes the output vocabulary token index ($i \in [1, \ldots, V]$, where V is the vocabulary size), and $z_{i,t}(\theta, S)$, $z_{i,t}(\hat{\theta}, S)$ represents the output logits for vocabulary token index i at suffix position t for the quantized ($\hat{\theta}$) and full precision (θ) models, respectively.

FLIP RATE under teacher forcing is basically an estimator for exact-match autoregressive verbatim regurgitation under greedy decoding. We formalize it as follows (proof in Appendix B):

Proposition 1 (Flip Rate computed under teacher forcing is an estimator of exact-match regurgitation under autoregressive greedy decoding)**.** *Let $\mathbb{I}[FR(S) = 0]$ be the indicator function where $\mathbb{I}[FR(S) = 0] = 1$ when $FR(S) = 0$ and $\mathbb{I}[FR(S) = 0] = 0$ when $FR(S) > 0$. Then, $\mathbb{I}[FR(S) = 0]$ computed under teacher-forcing indicates whether the quantized model will exactly regurgitate the ground truth suffix under autoregressive greedy decoding, given the original FP32 model exactly regurgitates S. Specifically,*

- *If $FR(S) = 0$, the quantized model exactly regurgitates the ground-truth suffix under autoregressive greedy decoding.*
- *If $FR(S) > 0$, the quantized model does not exactly regurgitate the ground-truth suffix under autoregressive greedy decoding.*

FLIP RATE is also a much cheaper metric than autoregressively generating suffix tokens to measure exact verbatim regurgitation. This is because teacher forcing allows all suffix positions to be evaluated in a single parallel forward pass, whereas autoregressive decoding requires sequential generation of the suffix tokens, with each newly generated token becoming the input for the next position.

To further understand how quantization precision affects verbatim regurgitation, we look into how quantization affects Flip Rate. We formalize this relation as follows (proof in Appendix C):

Theorem 1 (Upper Bound of Expected Flip Rate under Quantization)**.** *Let θ be an autoregressive full precision model. Then, under the assumption of uniform, symmetric quantization across parameters and Local Lipschitz continuity under model parameters, the upper bound of the expected flip rate ($UB(\mathbb{E}[FR], \theta, b)$) is as follows:*

$$UB(\mathbb{E}[FR], \theta, b) = \frac{1}{T} \sum_{t=1}^{T} F_{M_t}\left(\frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^b - 1}\right), \tag{3}$$

where t is the suffix position, T is the total number of evaluated suffix tokens, L_θ is the local Lipschitz constant, C_θ is the maximum absolute weight value of model θ, i.e., $max(|\theta_i|)$, b is the quantization precision in bits, FR is the random variable representing the flip rate of a sequence sampled from a data distribution $\mathcal{D}$, M_t is the random variable denoting the output logit margin at suffix position t for sequences sampled from $\mathcal{D}$, and $F_{M_t}(x) = \Pr(M_t \leq x)$ is the cumulative distribution function of M_t.

3.5 DATASETS AND MODELS

We conduct our experiments using the Pythia model suite(Biderman et al., 2022), which is trained on the publicly available Pile dataset (Gao et al., 2021). We utilize Pythia models in this work due to their open-source availability, and because prior work (Biderman et al., 2023) has released sets of training sequences that are extractable under the verbatim regurgitation definition discussed in Section 3.3. This makes Pythia a suitable testbed for systematically studying the impact of quantization on verbatim regurgitation. In this paper, we consider six models of sizes: 70M, 160M, 410M, 1.4B, 2.8B, and 6.9B parameters, trained on the deduplicated version of the Pile dataset. The publicly available regurgitated sequences by Pythia-models use (prefix,suffix) lengths of (32,32). We also experiment with varying lengths of (8,56), (16,48), (24,40), to show that our findings are independent of prefix length considered (Note: all figures consider lengths of (32,32) unless specified). To measure downstream task performance (test accuracy), we utilize 4 different datasets: ARC-Easy (Clark et al., 2018), Winogrande (Sakaguchi et al., 2019), Hellaswag (Zellers et al., 2019), & PIQA (Bisk et al., 2020) in zero-shot setting using the LLM evaluation framework (Gao et al., 2024).

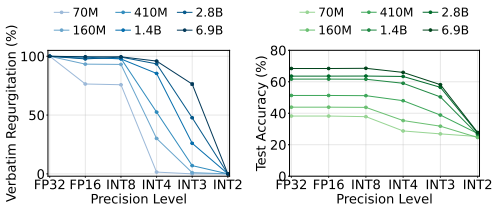


Figure 1: **Quantization reduces verbatim regurgitation as precision decreases.** As precision is reduced from FP32 to INT2, the regurgitation rate consistently declines, with higher precisions exhibiting greater regurgitation rate, across all model sizes. This reduction is accompanied by a drop in downstream test accuracy. Please note that the error bars are invisibly small (HQQ is used).

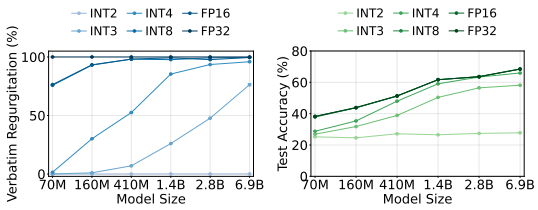


Figure 2: **Larger quantized models exhibit higher verbatim regurgitation than smaller ones.** After quantization, larger models show higher verbatim regurgitation rate than smaller ones when evaluated on a common set of sequences shared across all model sizes, ensuring a fair comparison, while also maintaining higher downstream test accuracy than smaller models. Please note that the error bars are invisibly small (HQQ is used).

An important point to note is that since we are comparing impact of quantization on verbatim regurgitation across different model sizes, we need to consider common set of sequences which are regurgitated by all six models for a fair comparison. Specifically, including sequences not regurgitated by all model sizes would confound the assessment of quantization effects and lead to biased comparisons. Hence, in this paper, for fair comparisons, we only consider the sequences that are commonly regurgitated (come out to be 338,440) by all six models in their FP32 baseline version. All of our experiments are performed across 3 random seeds for robustness in evaluation.

4 Impact of Quantization on Verbatim Regurgitation across Precision Levels, Model Size & Prefix-Suffix Length

In this section, we aim to understand how quantization impacts verbatim regurgitation across decreasing precision levels (FP32, FP16, INT8, INT4, INT3, and INT2), increasing model sizes (Pythia-70M, 160M, 410M, 1.4B, 2.8B, and 6.9B), varying (prefix,suffix) lengths, and multiple quantization techniques (HQQ, BitsAndBytes, and GPTQ). We present the results corresponding to these experiments in Sections 4.1, 4.2, 4.3, & Appendix E.1, respectively.

4.1 Precision Level: Verbatim Regurgitation Reduces with Decrease in Precision-Level

From Fig. 1, we observe that as the precision of model parameters decreases, the rate of verbatim regurgitation declines substantially. Specifically, it drops from 100% in FP32 to nearly 0% at INT2 precision. This trend indicates that quantization significantly affects the model's ability to store and reproduce exact training instances, with lower numerical precision progressively impairing this behavior. To further support this phenomenon, we carry out the flip rate analysis (see Section 3.4) and present them in Fig. 3a. We can observe that as the precision goes down, the flip rate increases, which essentially depicts that decreasing the precision reduces the model's tendency to regurgitate the train samples, as reflected in Fig. 1. We further establish an intuitive theoretical explanation as to why a decrease in precision increases the Flip-Rate. We utilize the upper bound of the Expected Flip Rate (UB$(\mathbb{E}[FR], \theta, b)$) obtained in Theorem 1, to establish the following corollary (proof in Appendix D):

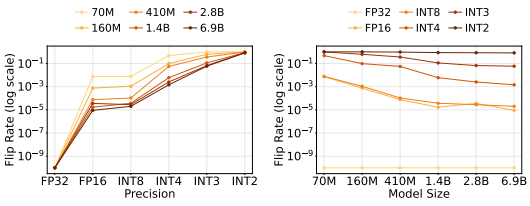
Corollary 1. *For a given model θ, upper bound of the expected flip rate (UB$(\mathbb{E}[FR], \theta, b)$) increases with decrease in model precision. Specifically, for two precision levels, b_1 and b_2 where $b1 > b2$:*

$$UB(\mathbb{E}[FR], \theta, b_1) \leq UB(\mathbb{E}[FR], \theta, b_2) \tag{4}$$

In addition to reduction in regurgitation, we also observe a decline in test accuracy (Figs. 1, 9) across all 4 downstream tasks, as the precision level decreases from FP32 to INT2. While prior studies (Frantar et al., 2022; Badri & Shaji, 2023; Esser et al., 2019) have reported accuracy degradation with decreasing precision, they have not examined its implications on regurgitation of training data.

4.2 Model Size: Quantized Larger Models Regurgitate More

In this section, we analyze the impact of model size on verbatim regurgitation under quantization. As described in Section 3.5, we construct a controlled comparison by considering the training sequences that are commonly regurgitated by all six Pythia models (i.e., the set of common sequences). This ensures a fair evaluation across different model sizes. Under this setup, Fig. 2 shows that, across all precision levels (excluding FP32 baseline), larger models consistently exhibit higher regurgitation rates than smaller models for the same set of sequences. This suggests that, even under reduced numerical precision, larger models retain sufficient effective capacity to store and reproduce more training instances than their smaller counterparts. To further understand this behavior, we analyze the flip rate across model sizes and present them in Fig. 3b. We can clearly observe that larger quantized models exhibit lower flip rate due to quantization in comparison to smaller models, correlating with our finding that for the same set of sequences larger quantized models regurgitate more than smaller ones, since fewer output tokens are flipped from the exact regurgitated version for larger quantized models. Note that this finding differs from prior works (Tirumala et al., 2022; Carlini et al., 2022), which primarily show that during training larger models (in full-precision settings) can store knowledge of a larger number of training sequences than smaller models, and can later regurgitate them more during inference. However, these studies do not explore the impact of quantization on verbatim regurgitation–which this paper investigates. One more key observation that we see from Fig. 2, is that regurgitation in smaller models starts to get impacted at higher precision levels, in comparison to larger models, which start to get impacted at lower precision levels. We also present the test-accuracy (for all 4 downstream tasks) of these quantized networks in Figs. 2, 9, reflecting how larger quantized networks perform better than their smaller counter-parts, an observation consistent with prior works (Esser et al., 2019; Li et al., 2021).



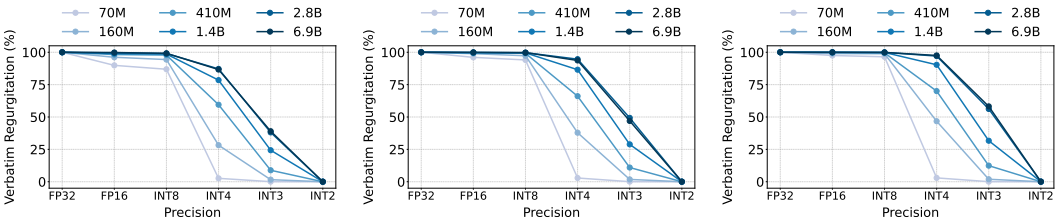
(a) Precision Level (b) Model Size

Figure 3: **Flip Rate analysis across precision levels and model sizes.** Flip Rate increases as precision decreases and is lower for larger quantized models than smaller ones. These observations provide support to the observed reduction in regurgitation at lower precisions (Section 4.1), and larger quantized models regurgitating more than smaller ones (Section 4.2) (HQQ is used).

4.3 Impact of Quantization on Regurgitation of Varying (Prefix,Suffix) Lengths

In this section, we verify whether our findings generalize to multiple (prefix,suffix) lengths. We consider: (8,56), (16,48), (24,40), apart from the original (32,32) prefix-suffix length setup. Since, we compare regurgitation across different (prefix,suffix) lengths, we first filter out the sequences which are commonly regurgitated under all 4 (prefix,suffix) length configurations of (8,56), (16,48), (24,40), & (32,32) by the FP32 versions of all models (Pythia-70M, 160M, 410M, 1.4B, 2.8B & 6.9B). We provide a detailed analysis of this filtration procedure in Appendix F. From Fig. 4 we observe that even for varying (prefix,suffix) lengths, verbatim regurgitation decreases with decrease in quantization precision, with larger quantized models regurgitating more than smaller ones.

Having established that, for a fixed (prefix,suffix) configuration, quantization consistently affects regurgitation across precision levels and model sizes, we next examine how the impact of quantization varies with the amount of prefix context. For each quantization precision (INT8, INT4, INT3, and INT2) and model size, we normalize the verbatim regurgitation rate at each (prefix,suffix) configuration by the corresponding rate for the (8,56) configuration. This provides a relative measure of how regurgitation changes as we increase the amount of prefix context. As shown in Fig. 5, among the controlled set of sequences that are commonly regurgitated at FP32 across all four (prefix,suffix) configurations, the regurgitation rate of quantized models increases with prefix length across quantization precisions and model sizes. This observation is also in part supported by findings in Carlini et al. (2022), who found that longer prefixes can increase the extractability of training data sequences, although we further examine the impact of prefix length under quantization. We also observe that for smaller models, the prefix length trend emerges at higher precisions and saturates at lower precisions. However, for larger models, the trend emerges at lower precisions. We attribute this



(a) (Prefix,Suffix) Length = (8,56) (b) (Prefix,Suffix) Length = (16,48) (c) (Prefix,Suffix) Length = (24,40)

Figure 4: **Consistent trends across varying (prefix,suffix) lengths.** Across different (prefix,suffix) lengths, verbatim regurgitation decreases with decrease in precision (Section 4.1) and larger quantized models regurgitate more than smaller ones (Section 4.2) (HQQ is used).

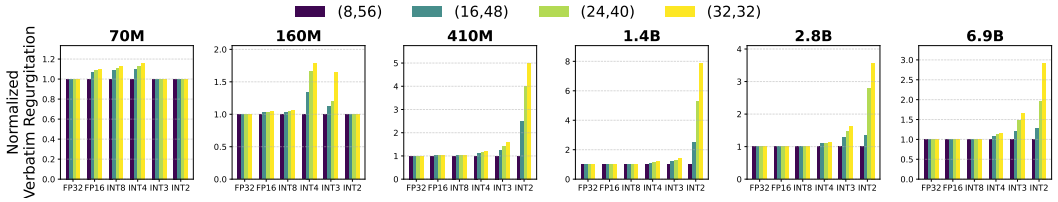


Figure 5: **Impact of quantization on regurgitation across varying (prefix,suffix) lengths.** Among the common set of sequences that are regurgitated under all (prefix,suffix) lengths at FP32 precision, we measure the relative normalized regurgitation between a fixed (prefix,suffix) length and the shortest prefix length configuration of (8,56) under quantization. We observe that regurgitation increases as the prefix length increases under quantization, with additional discussions across model sizes provided in Section 4.3 (HQQ is used).

to the observations made in Section 4.2: for larger models, drop in regurgitation emerges at lower precisions, whereas for smaller models it starts at higher precision and saturates at lower precisions.

5 Scaling Law for Verbatim Regurgitation under Quantization

In this section, we propose a scaling law capturing how verbatim regurgitation behaves under quantization, across three axes: quantization precision, model size, and prefix-length. In this study, we consider 6 different precisions (FP32, FP16, INT8, INT4, INT3, and INT2), 6 model sizes (70M, 160M, 410M, 1.4B, 2.8B, and 6.9B), and 4 prefix-lengths (8, 16, 24, and 32) to derive the scaling law. To formulate the scaling law for regurgitation under quantization, we utilize the observation made in Fig. 1 where verbatim regurgitation behaves as an S-shaped function across model precisions, with its inflection point (when regurgitation begins to drop from 100%) moving towards lower precision, with the S-shaped function also becoming steeper as model size increases. Furthermore, we also observe in Fig. 4 that as the prefix length increases, regurgitation also increases under quantization. Hence, based on these trends, we propose the following scaling law, $\mathrm{VR}(b, N, L)$:

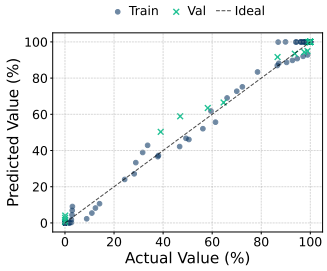
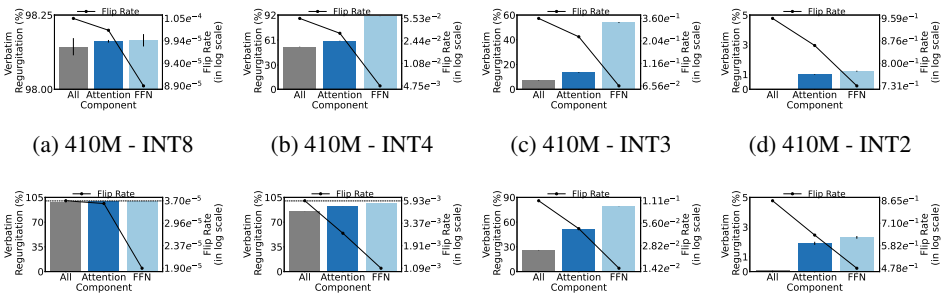


Figure 6: **Scaling Law for Verbatim Regurgitation under Quantization.** 6.9B is the held-out model with the rest of the models used for fitting across 6 precisions and 4 prefix lengths.

$$\mathrm{VR}(b, N, L) = 100 \cdot \exp\left(-\exp\left(-c_1\left(b - \left(c_2 - \frac{c_3}{1+\exp\left(-c_4(\log N - \log c_5)\right)} - c_6 \log L\right)\right)\right)\right) \tag{5}$$

where, b represents quantization precision in bits, N represents model size, L refers to prefix length, exp is the exponential function, and $c_1, c_2, c_3, c_4, c_5, c_6$ are the parameters to be estimated. We also provide analysis across 4 different scaling law equations (including ours), and the rationale behind the scaling laws in Appendix G.1. From Fig. 6, we observe that our proposed scaling law fits well



(a) 410M - INT8 (b) 410M - INT4 (c) 410M - INT3 (d) 410M - INT2

(e) 1.4B - INT8 (f) 1.4B - INT4 (g) 1.4B - INT3 (h) 1.4B - INT2

Figure 7: **Quantizing Attention module has a larger impact on verbatim regurgitation than FFN.** Quantizing attention modules leads to a higher reduction in regurgitation compared to FFN, correlating with the higher flip rate (depicted by the black line) corresponding to attention modules quantization than the FFN modules (HQQ is used).

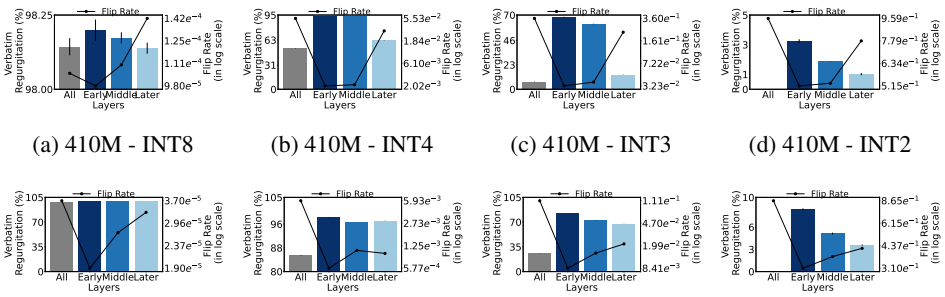
with a train R^2-score = 0.995 and Mean Absolute Error (MAE) = 1.61%, alongside a validation R^2-score = 0.988 and MAE = 2.20% for the held-out 6.9B model. The estimated parameters for this fit are $c_1 = 3.0713, c_2 = 10.0000, c_3 = 7.0143, c_4 = 0.4485, c_5 = 0.0142$, and $c_6 = 0.1878$. We also provide a cross-validation analysis across other remaining models (2.8B, 1.4B, 410M, 160M, and 70M), precision levels (FP32, FP16, INT8, INT4, INT3, and INT2), and prefix lengths (8, 16, 24, and 32) in Appendix G.2 & Appendix G.3.

6 Does Quantizing Specific Components and Layers has More Impact on Verbatim Regurgitation?

We now investigate whether quantizing specific components and/or layers in the network has a more pronounced impact on verbatim regurgitation. We divide our analysis into two sections, studying the impact of quantization across: (i) components: attention and feed-forward network modules (Section 6.1), and (ii) layers: early, middle and later (Section 6.2). Please note that we do not explore the impact of quantizing LayerNorm component on verbatim regurgitation in this paper. It is because prior works (Li et al., 2024; Prato et al., 2020) have shown that quantizing LayerNorm significantly degrades model performance. Accordingly, almost all quantization approaches avoid quantizing LayerNorm by keeping it in sufficiently high precision levels. Additionally, for the experiments corresponding to layer-wise and component-wise quantization, we do not study FP16, as models in FP16 precision have almost identical regurgitation rate and test accuracy to the FP32 baseline models, as seen from Figs. 1 & 2. Lastly, we also compute the Flip Rate metric to provide further support to our analyses.

6.1 Attention Module Quantization has Higher Impact on Verbatim Regurgitation than Feed-Forward Network (FFN)

In this section, we examine whether quantizing specific components in LLMs leads to differing impacts on verbatim regurgitation. As explained earlier, attention and FFN modules are the key components in current modern LLMs, hence we study the impact of quantizing them on regurgitation. Specifically, we compare quantizing only attention modules (all their sublayers: query, key, value, and output projections) against quantizing only the feed-forward network (FFN) modules (all their sublayers: up and down projections), across the whole network. As shown in Fig. 7, 15, although the gaps are not significant for all instances (refer to Appendix J for additional discussion), a consistent trend is observed across all model sizes (Pythia-70M, 160M, 410M, and 1.4B) and precision levels (INT8, INT4, INT3, and INT2): quantizing the attention modules leads to a larger reduction in verbatim regurgitation compared to quantizing the FFN modules alone, suggesting the attention module has more impact on regurgitation than FFN. These trends in verbatim regurgitation is also reflected in the Flip Rate metric (depicted by the black line) where quantizing attention modules cause more increase in Flip Rate than quantizing only FFN. In addition to regurgitation, we also



(a) 410M - INT8 (b) 410M - INT4 (c) 410M - INT3 (d) 410M - INT2

(e) 1.4B - INT8 (f) 1.4B - INT4 (g) 1.4B - INT3 (h) 1.4B - INT2

Figure 8: **Later layers quantization has a larger impact on verbatim regurgitation.** Quantizing later layers leads to a higher reduction in regurgitation compared to early and middle layers. This trend correlates further with the higher flip rate for later layers quantization (depicted by the black line) in comparison to early-middle layers (HQQ is used).

compare the impact of attention and FFN components quantization on test accuracy in Appendix K, where we observe that, for ARC-E dataset, attention modules quantization causes a relatively higher drop in test accuracy than FFN modules quantization, but for other 3 datasets–Hellaswag, PIQA & Winogrande, quantizing attention and FFN modules separately results in a broadly similar impact on test accuracy across model sizes and precision levels.

6.2 Later Layers Quantization have More Impact on Verbatim Regurgitation than Early-Middle Layers

Beyond component-wise analysis, we also examine whether quantizing specific layers in LLMs leads to differing impacts on verbatim regurgitation. To this end, we partition an N-layer network into three regions—early, middle, and later layers—following (Singhal & Kim, 2025). Layers 1 to $\lfloor N/3 \rfloor$ correspond to the *early layers* , layers $\lfloor N/3 \rfloor + 1$ to $\lfloor 2N/3 \rfloor$ to the *middle layers*, and the remaining layers from $\lfloor 2N/3 \rfloor + 1$ to N to the *later layers*. We compare the effect of quantizing each layer's region independently (we quantize both attention and FFN modules in each region), as well as quantizing all layers at once. As shown in Fig. 8, 16 although the gaps are not significant for all instances (refer to Appendix J for additional discussion), a consistent trend emerges across model sizes (Pythia-70M, 160M, 410M, and 1.4B) and precision levels (INT8, INT4, INT3, and INT2): quantizing the later layers leads to a larger reduction in verbatim regurgitation relative to quantizing the early-middle layers, suggesting importance of later layers in verbatim regurgitation of training sequences. These trends in verbatim regurgitation are also reflected in the Flip Rate metric (depicted by the black line), where quantizing later layers cause more increase in Flip Rate than quantizing early or middle layers. In addition to regurgitation, we also analyze the impact of quantization across these layer groups on test accuracy, as presented in Appendix L, where we observe that quantizing early, middle, and later layers separately results in a broadly similar impact on test accuracy across model sizes and precision levels, for all 4 downstream tasks.

7 Conclusion

In conclusion, our work provides the first systematic investigation of the impact of post-training quantization on verbatim regurgitation in language models. Our analysis indicates that reducing numerical precision results in a decrease in verbatim regurgitation and larger quantized models exhibit higher regurgitation rates than smaller ones, consistent across varying prefix & suffix lengths. We also propose a scaling law for modeling verbatim regurgitation under quantization. Furthermore, a fine-grained analysis across layers and components reveals consistent patterns: quantizing later layers leads to a larger reduction in regurgitation than early and middle layers, and quantizing attention modules has a higher effect than feed-forward network, with all the trends well captured by the Flip Rate metric analysis. Overall, through this paper, we aim to better understand the impact of quantization on regurgitation. We discuss the limitations of our work in Appendix M.

AI USE STATEMENT

In this work, we used generative AI tools for coding related tasks, refining theoretical formulations, and polishing wordings in the text. We have not used generative AI tools for ideation or experimental setups. The authors have reviewed all AI-assisted parts and take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We utilize 1xH100 GPU for quantizing and evaluating Pythia-2.8B and 6.9B models, and 1xA5000 and 1xRTX4060 GPUs for other remaining models. Our code is available as a supplementary file.

REFERENCES

K Alizadeh, I Mirzadeh, D Belenko, K Khatamifard, M Cho, CC Del Mundo, M Rastegari, and M Farajtabar. Llm in a flash: Efficient large language model inference with limited memory, arxiv. *arXiv preprint arXiv:2312.11514*, 2024.

Hicham Badri and Appu Shaji. Half-quadratic quantization of large machine learning models, November 2023. URL `https://dropbox.github.io/hqq_blog/`.

Abhimanyu Bellam and Jung-Eun Kim. Explaining how quantization disparately skews a model. *arXiv preprint arXiv:2509.07222*, 2025.

Stella Biderman, Quentin Anthony Hailey Schoelkopf, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Edward Raff USVSN Sai Prashanth, Aviya Skowron, Lintang Sutawika, et al. Pythia: A suite for analyzing large language models across training and scaling, 2023. *URL https://arxiv. org/abs/2304.01373*, 2304, 2022.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. Emergent and predictable memorization in large language models. *Advances in Neural Information Processing Systems*, 36:28072–28090, 2023.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Nicolas Boizard, Kevin El Haddad, Céline Hudelot, and Pierre Colombo. Towards cross-tokenizer distillation: the universal logit distillation loss for llms. *arXiv preprint arXiv:2402.12030*, 2024.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. *Advances in Neural Information Processing Systems*, 36:75067–75096, 2023.

Jaydeep Borkar, Karan Chadha, Niloofar Mireshghallah, Yuchen Zhang, Irina-Elena Veliche, Archi Mitra, David A Smith, Zheng Xu, and Diego Garcia-Olano. Memorization dynamics in knowledge distillation for language models. *arXiv preprint arXiv:2601.15394*, 2026.

Thierry Bossy, Julien Vignoud, Tahseen Rabbani, Juan R Troncoso Pastoriza, and Martin Jaggi. Mitigating unintended memorization with lora in federated learning for llms. *arXiv preprint arXiv:2502.05087*, 2025.

Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. What does it mean for a language model to preserve privacy? In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pp. 2280–2292, 2022.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Paper review:'sparks of artificial general intelligence: Early experiments with gpt-4'. 2023.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2022.

Ting-Yun Chang, Jesse Thomason, and Robin Jia. Do localization methods actually localize memorized data in llms? a tale of two benchmarks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3190–3211, 2024.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. Efficientqat: Efficient quantization-aware training for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10081–10100, 2025.

Ting-An Chen, De-Nian Yang, and Ming-Syan Chen. Overcoming forgetting catastrophe in quantization-aware training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17358–17367, 2023.

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.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*, 2022.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 1286–1305, 2021.

Abhinav Dutta, Sanjeev Krishnan, Nipun Kwatra, and Ramachandran Ramjee. Accuracy is not all you need. *Advances in Neural Information Processing Systems*, 37:124347–124390, 2024.

Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

L Gao, S Biderman, S Black, L Golding, T Hoppe, C Foster, J Phang, H He, A Thite, N Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling, 2020. *arXiv preprint arXiv:2101.00027*, 2021.

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, 07 2024. URL `https://zenodo.org/records/12608602`.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-power computer vision*, pp. 291–326. Chapman and Hall/CRC, 2022.

Benjamin Gompertz. On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. In *Abstracts of the Papers Printed in the Philosophical Transactions of the Royal Society of London*, volume 2, pp. 252–253. JSTOR, 1815.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Junyuan Hong, Jinhao Duan, Chenhui Zhang, Zhangheng Li, Chulin Xie, Kelsey Lieberman, James Diffenderfer, Brian Bartoldson, Ajay Jaiswal, Kaidi Xu, et al. Decoding compressed trust: Scrutinizing the trustworthiness of efficient llms under compression. *arXiv preprint arXiv:2403.15447*, 2024.

Jing Huang, Diyi Yang, and Christopher Potts. Demystifying verbatim memorization in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10711–10732, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Jason Kong, Nilesh Prasad Pandey, Flavio Ponzina, and Tajana Rosing. A kl lens on quantization: Fast, forward-only sensitivity for mixed-precision ssm-transformer models. *arXiv preprint arXiv:2604.13440*, 2026.

Andrey Kuzmin, Markus Nagel, Mart Van Baalen, Arash Behboodi, and Tijmen Blankevoort. Pruning vs quantization: Which is better? *Advances in neural information processing systems*, 36:62414–62427, 2023.

Liang Li, Qingyuan Li, Bo Zhang, and Xiangxiang Chu. Norm tweaking: High-performance low-bit quantization of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18536–18544, 2024.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*, 2021.

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. *Proceedings of machine learning and systems*, 6: 87–100, 2024.

Bing Liu, Wenjun Miao, Boyu Zhang, Qiankun Zhang, Bin Yuan, Jing Wang, Shenghao Liu, and Xianjun Deng. Understanding the unfairness in network quantization. In *Forty-second International Conference on Machine Learning*, 2025.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 467–484, 2024.

Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 346–363. IEEE, 2023.

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart Van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.

Chenjie Ni, Zhepeng Wang, Runxue Bao, Shangqian Gao, and Yanfu Zhang. Controllable memorization in llms via weight pruning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 15141–15156, 2025.

Gabriele Prato, Ella Charlaix, and Mehdi Rezagholizadeh. Fully quantized transformer for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1–14, 2020.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

Rishi Singhal and Jung-Eun Kim. Impact of layer norm on memorization and generalization in transformers. *arXiv preprint arXiv:2511.10566*, 2025.

Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. Beyond memorization: Violating privacy via inference with large language models. *arXiv preprint arXiv:2310.07298*, 2023.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022.

Fei Wang and Baochun Li. Leaner training, lower leakage: Revisiting memorization in llm fine-tuning with lora. *arXiv preprint arXiv:2506.20856*, 2025.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pp. 38087–38099. PMLR, 2023.

Z Yao, RY Aminabadi, M Zhang, X Wu, C Li, and Y Zeroquant He. Efficient and affordable post-training quantization for large-scale transformers, 2022. *URL https://arxiv. org/abs/2206.01861*, 2022.

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*, 2019.

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. Catastrophic failure of llm unlearning via quantization. *arXiv preprint arXiv:2410.16454*, 2024.

Jiaqi Zhao, Ming Wang, Miao Zhang, Yuzhang Shang, Xuebo Liu, Yaowei Wang, Min Zhang, and Liqiang Nie. Benchmarking post-training quantization in llms: Comprehensive taxonomy, unified evaluation, and comparative analysis. *arXiv preprint arXiv:2502.13178*, 2025.

APPENDIX

A ISSUES IN USING SQNR AND KL-DIVERGENCE METRICS AS A PROXY FOR VERBATIM REGURGITATION

Both SQNR and KL-divergence metrics were originally introduced to study how quantization impacts the output logits compared to the baseline FP32 model. However, a recent work (Kong et al., 2026) showed how KL-divergence is a tighter bound than SQNR to measure the impact of quantization on the output logits distribution. Hence, we rule out SQNR as an explanation metric for our study in the first place.

Regarding KL-divergence, even though it provides a tighter bound than SQNR to measure the impact on output logits under quantization, it still fails short to capture the discrete nature of token-level regurgitation, as explained below:

Counter-Example Showing Failure to use KL-divergence as a Proxy for Verbatim Regurgitation:

Consider a verbatim regurgitated sequence a with prefix p and suffix s, such that $a = p + s$. We evaluate this sequence under a teacher-forcing setting using both the original FP32 model and quantized models, each defined over a vocabulary of size V. At a given suffix position, let the models output the following probability distributions over the vocabulary, with the vocabulary token x being the ground truth regurgitated token for that suffix position:

$$P_{\text{FP32}} = [p_1^{\text{FP32}}, \ldots, p_x^{\text{FP32}}, \ldots, p_V^{\text{FP32}}]$$
$$P_{\text{Q}} = [p_1^{\text{Q}}, \ldots, p_x^{\text{Q}}, \ldots, p_V^{\text{Q}}],$$

where p_i^{FP32} & p_i^{Q} represent the probability of predicting vocabulary token i corresponding to the FP32 and the quantized models, respectively.

For simple illustration purposes, consider $V = 4$, where the third token in the vocabulary corresponds to the ground-truth suffix token. Then, let us suppose that quantization disrupts the probability distribution as the following example:

$$P_{\text{FP32}} = [0.16, 0.17, \mathbf{0.50}, 0.17]$$
$$P_{\text{Q}} = [0.09, 0.06, \mathbf{0.45}, 0.40]$$

Here, the ground-truth regurgitated token corresponds to index $x = 3$, where $p_3^{\text{FP32}} = 0.50$ and $p_3^{\text{Q}} = 0.45$. Consequently, both models assign the same top-1 prediction under greedy decoding, and the generated suffix token remains identical. Therefore, from the perspective of verbatim regurgitation, the behavior is preserved.

However, in the example, KL-divergence between P_{FP32} and P_{Q} is large, i.e., based on the KL-divergence formula:

$$\text{D}_{\text{KL}}(P_{\text{FP32}} \,\|\, P_{\text{Q}}) = \sum_{i=1}^{4} p_i^{\text{FP32}} \ln \frac{p_i^{\text{FP32}}}{p_i^{\text{Q}}}$$

For the example above:

$$\text{D}_{\text{KL}}(P_{\text{FP32}} \,\|\, P_{\text{Q}}) = 0.16 \ln \frac{0.16}{0.09} + 0.17 \ln \frac{0.17}{0.06} + 0.50 \ln \frac{0.50}{0.45} + 0.17 \ln \frac{0.17}{0.40} = 0.175 \text{ nats}$$

This serves as a counterexample where KL-divergence indicates a significant shift in the overall probability distribution, despite no change in the actual regurgitated token. Hence, showing that KL-divergence is not a faithful (guaranteed) proxy for verbatim regurgitation.

B PROOF OF PROPOSITION 1

Proof. Let $Y = (y_1, y_2, \ldots, y_T)$ be the ground truth suffix (given that the original FP32 model exactly regurgitates), $\hat{Y}^{\text{AR}} = (\hat{y}_1^{\text{AR}}, \hat{y}_2^{\text{AR}}, \ldots, \hat{y}_T^{\text{AR}})$ denotes the sequence generated by the quantized model $\hat{\theta}$ under greedy autoregressive (AR) decoding given prefix X, and $\hat{Y}^{\text{TF}} = (\hat{y}_1^{\text{TF}}, \hat{y}_2^{\text{TF}}, \ldots, \hat{y}_T^{\text{TF}})$ represents the sequence generated by the quantized model $\hat{\theta}$ under greedy teacher-forcing (TF) decoding given prefix X. We prove the 2 cases separately as follows:

Case 1: When $FR(S) = 0$

Since

$$FR(S) = \frac{1}{T}\sum_{t=1}^{T} I_t(S) = 0, \tag{6}$$

and $I_t(S) \in \{0, 1\}$, therefore to satisfy equation 6

$$I_t(S) = 0, \qquad \forall t \in \{1, \ldots, T\} \tag{7}$$

Thus, under teacher forcing,

$$\hat{y}_t^{\mathrm{TF}} = y_t, \qquad \forall t \in \{1, \ldots, T\} \tag{8}$$

To show that when $FR(S) = 0$, the quantized model exactly regurgitates the ground truth suffix Y under autoregressive greedy decoding, we prove that

$$\hat{y}_t^{\mathrm{AR}} = \hat{y}_t^{\mathrm{TF}} = y_t, \qquad \forall t \in \{1, \ldots, T\}, \tag{9}$$

by induction over the suffix positions.

Base case ($t = 1$). Both teacher forcing and autoregressive decoding receive the same prefix X. Therefore, under greedy decoding,

$$\hat{y}_1^{\mathrm{AR}} = \arg\max p_{\hat{\theta}}(. \mid X) = \hat{y}_1^{\mathrm{TF}} = y_1 \tag{10}$$

Inductive step. Suppose that $(\hat{y}_1^{\mathrm{AR}}, \hat{y}_2^{\mathrm{AR}}, \ldots, \hat{y}_{t-1}^{\mathrm{AR}}) = (\hat{y}_1^{\mathrm{TF}}, \hat{y}_2^{\mathrm{TF}}, \ldots, \hat{y}_{t-1}^{\mathrm{TF}}) = (y_1, y_2, \ldots, y_{t-1})$, i.e.,

$$\hat{y}_{<t}^{\mathrm{AR}} = \hat{y}_{<t}^{\mathrm{TF}} = y_{<t} \tag{11}$$

Then the autoregressive and teacher-forced contexts to predict at suffix position t are identical:

$$(X, \hat{y}_{<t}^{\mathrm{AR}}) = (X, \hat{y}_{<t}^{\mathrm{TF}}) = (X, y_{<t}) \tag{12}$$

Since both predictions are generated by the same quantized model $\hat{\theta}$ using greedy decoding,

$$\hat{y}_t^{\mathrm{AR}} = \hat{y}_t^{\mathrm{TF}} = y_t \tag{13}$$

Therefore, by induction,

$$\hat{Y}^{\mathrm{AR}} = \hat{Y}^{\mathrm{TF}} = Y \tag{14}$$

Case 2: When $FR(S) > 0$

Since $FR(S) > 0$, there exists at least one suffix position at which the quantized model flips its prediction under teacher forcing. Hence, we first define the suffix position at which the first teacher-forced flip happens:

$$t^\star = \min\{t : I_t(S) = 1\} \tag{15}$$

By definition of $t^\star$,

$$I_i(S) = 0, \qquad \forall i < t^\star \tag{16}$$

and therefore

$$\hat{y}_i^{\mathrm{TF}} = y_i, \qquad \forall i < t^\star \tag{17}$$

From Case 1 proof, we can infer that as long as the teacher-forced prediction remains the same as the ground truth suffix token, the greedy autoregressive prediction will also remain the same. Therefore, before $t^\star$ suffix position where the first teacher-forced flip happens, the autoregressive and teacher-forced predictions remain identical to the ground truth suffix tokens:

$$\hat{y}_{<t^\star}^{\mathrm{AR}} = \hat{y}_{<t^\star}^{\mathrm{TF}} = y_{<t^\star} \tag{18}$$

We know from equation 15 that the first teacher-forced flip happens at suffix position $t^\star$. Hence, $I_{t^\star}(S) = 1$, which implies

$$\hat{y}^{\mathrm{TF}}_{t^\star} \neq y_{t^\star} \tag{19}$$

From equation 18, we also know that the autoregressive context to make a prediction at suffix position $t^\star$ is exactly the same as the teacher-forced context:

$$(X, \hat{y}^{\mathrm{AR}}_{<t^\star}) = (X, \hat{y}^{\mathrm{TF}}_{<t^\star}) = (X, y_{<t^\star}) \tag{20}$$

Since both the teacher-forcing and autoregressive predictions are done by the same quantized model $\hat{\theta}$ under the same prefix context and greedy decoding. Hence, combining it with equation 19, we obtain:

$$\hat{y}^{\mathrm{AR}}_{t^\star} = \hat{y}^{\mathrm{TF}}_{t^\star} \neq y_{t^\star} \tag{21}$$

Thus, the autoregressive output differs from the ground-truth suffix at position $t^\star$. This essentially means, when $FR(S) > 0$, the quantized model fails to exactly regurgitate the ground truth suffix under greedy autoregressive decoding, i.e., $\hat{Y}^{\mathrm{AR}} \neq Y$.

Therefore, considering both Case 1 and Case 2, we can infer that $\mathbb{I}[FR(S) = 0]$ computed under teacher-forcing directly indicates whether the quantized model will exactly regurgitate the ground truth suffix under autoregressive greedy decoding, where:

- If $FR(S) = 0$, the quantized model exactly regurgitates the ground-truth suffix under autoregressive greedy decoding.
- If $FR(S) > 0$, the quantized model does not exactly regurgitate the ground-truth suffix under autoregressive greedy decoding.

□

C Proof of Theorem 1

Proof. Let θ denote the full-precision model, and $\Delta\theta$ denotes the induced quantization error. Then the quantized model $\hat{\theta}$ can be written as:

$$\hat{\theta} = \theta + \Delta\theta \tag{22}$$

Under uniform symmetric quantization with b bits, in the parameter range $[-C_\theta, C_\theta]$, the step size is $\Delta = 2C_\theta/(2^b - 1)$. Since each parameter is rounded to the nearest grid point, the quantization error for each parameter satisfies $|\Delta\theta_n| \leq \Delta/2$. Hence, the absolute quantization error of each parameter is bounded by:

$$|\Delta\theta_n| \leq \frac{2C_\theta}{2 \cdot (2^b - 1)} = \frac{C_\theta}{2^b - 1}, \qquad n = 1, \ldots, |\theta| \tag{23}$$

Therefore, the whole model's upper bound of parameter perturbation can be derived as:

$$\|\Delta\theta\|_2 = \sqrt{\sum_{n=1}^{|\theta|} |\Delta\theta_n|^2} \leq \sqrt{\frac{C_\theta^2 |\theta|}{(2^b - 1)^2}} = \frac{C_\theta \sqrt{|\theta|}}{2^b - 1} \tag{24}$$

Similarly, quantization error of the model logits for vocabulary token at index $i \in [1, ...V]$ (V is the output vocabulary size) at suffix position t for a sequence S (sampled from a data distribution $\mathcal{D}$) can be written as:

$$e_{i,t}(S) = z_{i,t}(\hat{\theta}, S) - z_{i,t}(\theta, S), \quad \forall i \in [1, ..., V] \tag{25}$$

where $z_{i,t}(\hat{\theta})$ and $z_{i,t}(\theta)$ denote the output logits for vocabulary token index i at suffix position t for the quantized ($\hat{\theta}$) and full precision (θ) models, respectively.

Now assume that the model logits at suffix position t for every sequence S in $\mathcal{D}$ are locally Lipschitz with respect to the model parameters. In particular, there exists a constant $L_\theta \geq 0$ such that

$$|z_{i,t}(\hat{\theta}, S) - z_{i,t}(\theta, S)| \leq L_\theta \|\hat{\theta} - \theta\|_2, \quad \forall i \in [1, ..., V] \tag{26}$$

Combining equation 22, equation 24, equation 25, and equation 26, we obtain

$$|e_{i,t}(S)| = |z_{i,t}(\hat{\theta}, S) - z_{i,t}(\theta, S)| \leq \frac{L_\theta C_\theta \sqrt{|\theta|}}{2^b - 1} \tag{27}$$

Now, let

$$i^\star = \arg\max_i z_{i,t}(\theta, S) \tag{28}$$

denote the predicted token index by the full-precision model θ at position t for sequence S.

Using equation 28, we can write the output margin for the full-precision model at position t for sequence S, $M_t(S)$, as follows:

$$M_t(S) = z_{i^\star,t}(\theta, S) - \max_{k \neq i^\star} z_{k,t}(\theta, S) \tag{29}$$

Since S is sampled from $\mathcal{D}$, $M_t(S)$ varies across evaluation sequences. Thus, for each fixed suffix position t, we regard

$$M_t := M_t(S), \qquad S \sim \mathcal{D}, \tag{30}$$

as a random variable.

By equation 25, we obtain that:

$$z_{i,t}(\hat{\theta}, S) = z_{i,t}(\theta, S) + e_{i,t}(S) \tag{31}$$

Suppose that a prediction flip occurs at position t, where quantization changes the model prediction from the top token at index $i^\star$ in logits to the second-highest token at index k, then

$$z_{k,t}(\hat{\theta}, S) \geq z_{i^\star,t}(\hat{\theta}, S) \tag{32}$$

By equation 31, the flip condition in equation 32 becomes

$$z_{k,t}(\theta, S) + e_{k,t}(S) \geq z_{i^\star,t}(\theta, S) + e_{i^\star,t}(S) \tag{33}$$

Rearranging equation 33 gives

$$z_{i^\star,t}(\theta, S) - z_{k,t}(\theta, S) \leq e_{k,t}(S) - e_{i^\star,t}(S) \tag{34}$$

By the definition of $M_t(S) = z_{i^\star,t}(\theta, S) - z_{k,t}(\theta, S)$ in equation 29, we obtain the necessary condition for flipping to happen at position t:

$$M_t(S) \leq e_{k,t}(S) - e_{i^\star,t}(S) \tag{35}$$

For the sake of simplicity, let $\frac{L_\theta C_\theta \sqrt{|\theta|}}{2^b - 1} = \delta$ in equation 27. Hence, each individual output logit perturbation is rewritten as:

$$|e_{i,t}(S)| = |z_{i,t}(\hat{\theta}, S) - z_{i,t}(\theta, S)| \leq \delta \tag{36}$$

The largest possible relative perturbation between two competing logits occurs when

$$e_{k,t}(S) = +\delta \qquad \text{and} \qquad e_{i^\star,t}(S) = -\delta \tag{37}$$

Consequently, equation 35 implies

$$M_t(S) \leq e_{k,t}(S) - e_{i^\star,t}(S) \leq \delta - (-\delta) = 2\delta \tag{38}$$

Therefore, by equation 38, a prediction flip at suffix position t can occur only if the original logit margin $M_t(S)$ is no larger than 2δ. Equivalently, every sequence that experiences a prediction flip at position t must belong to the set of sequences satisfying $M_t(S) \leq 2\delta$.

To formalize this relation among the two sets, we define the event set of sequences in $\mathcal{D}$ that encounter a prediction flip at position t:

$$\text{Flip}_t = \left\{ S : \arg\max_i z_{i,t}(\theta, S) \neq \arg\max_i z_{i,t}(\hat{\theta}, S) \right\} \tag{39}$$

As explained above, Flip_t event set satisfies the following inclusion:

$$\text{Flip}_t \subseteq \{ S : M_t(S) \leq 2\delta \} \tag{40}$$

Taking probabilities with respect to $S \sim \mathcal{D}$ on both sides of the event inclusion in equation 40 with equation 30 gives

$$\Pr_{S\sim\mathcal{D}}(S \in \text{Flip}_t) \leq \Pr_{S\sim\mathcal{D}}(M_t(S) \leq 2\delta) = \Pr(M_t \leq 2\delta) \tag{41}$$

Let F_{M_t} denote the cumulative distribution function (CDF) of the random variable M_t. By definition,

$$F_{M_t}(x) = \Pr(M_t \leq x). \tag{42}$$

Therefore, evaluating equation 42 at $x = 2\delta$ gives

$$F_{M_t}(2\delta) = \Pr(M_t \leq 2\delta). \tag{43}$$

Combining equation 41 and equation 43, we obtain

$$\Pr_{S\sim\mathcal{D}}(S \in \text{Flip}_t) \leq F_{M_t}(2\delta) \tag{44}$$

From equation 2, we know that the Flip-Rate for a sequence S, $FR(S)$ is written as:

$$FR(S) = \frac{1}{T}\sum_{t=1}^{T} I_t(S) \tag{45}$$

where,

$$I_t(S) = \mathbb{I}\left[\arg\max_i z_{i,t}(\theta, S) \neq \arg\max_i z_{i,t}(\hat{\theta}, S)\right] = \begin{cases} 1, & \text{if } \arg\max_i z_{i,t}(\theta, S) \neq \arg\max_i z_{i,t}(\hat{\theta}, S) \\ 0, & \text{if } \arg\max_i z_{i,t}(\theta, S) = \arg\max_i z_{i,t}(\hat{\theta}, S) \end{cases} \tag{46}$$

Taking expectations of equation 45 and using linearity of expectation gives

$$\begin{aligned} \mathbb{E}[FR] &= \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} I_t(S)\right] \\ &= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{S\sim\mathcal{D}}[I_t(S)] \\ &= \frac{1}{T}\sum_{t=1}^{T} 1 \cdot \Pr_{S\sim\mathcal{D}}\left(\arg\max_i z_{i,t}(\theta, S) \neq \arg\max_i z_{i,t}(\hat{\theta}, S)\right) \\ &\qquad + 0 \cdot \Pr_{S\sim\mathcal{D}}\left(\arg\max_i z_{i,t}(\theta, S) = \arg\max_i z_{i,t}(\hat{\theta}, S)\right) \\ &= \frac{1}{T}\sum_{t=1}^{T} \Pr_{S\sim\mathcal{D}}\left(\arg\max_i z_{i,t}(\theta, S) \neq \arg\max_i z_{i,t}(\hat{\theta}, S)\right) \end{aligned} \tag{47}$$

From equation 39, we can substitute $\Pr_{S\sim\mathcal{D}}\left(\arg\max_i z_{i,t}(\theta, S) \neq \arg\max_i z_{i,t}(\hat{\theta}, S)\right) = \Pr_{S\sim\mathcal{D}}(S \in \text{Flip}_t)$ into equation 47, to obtain:

$$\mathbb{E}[FR] = \frac{1}{T}\sum_{t=1}^{T} \Pr_{S\sim\mathcal{D}}(S \in \text{Flip}_t) \tag{48}$$

From equation 44, we know that $\Pr_{S\sim\mathcal{D}}(S \in \text{Flip}_t) \leq F_{M_t}(2\delta)$. Hence, applying it in equation 48, we obtain

$$\mathbb{E}[FR] \leq \frac{1}{T}\sum_{t=1}^{T} F_{M_t}(2\delta) \tag{49}$$

Substituting $\delta = \frac{L_\theta C_\theta \sqrt{|\theta|}}{2^b - 1}$ into equation 49 yields the upper bound of the expected Flip Rate $\text{UB}(\mathbb{E}[FR])$:

$$\boxed{\text{UB}(\mathbb{E}[FR], \theta, b) = \frac{1}{T}\sum_{t=1}^{T} F_{M_t}\left(\frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^b - 1}\right)} \tag{50}$$

□

D Proof of Corollary 1

Proof. From Theorem 1, we know that the upper bound of the expected flip rate ($\text{UB}(\mathbb{E}[FR], \theta, b)$) is:

$$\text{UB}(\mathbb{E}[FR], \theta, b) = \frac{1}{T}\sum_{t=1}^{T} F_{M_t}\left(\frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^b - 1}\right) \tag{51}$$

Since we compute the upper bound for the same model θ at two different precision levels b_1 and b_2 (where $b_1 > b_2$), we can write the upper bound at b_1 and b_2 precision levels, respectively, as follows:

$$\text{UB}(\mathbb{E}[FR], \theta, b_1) = \frac{1}{T}\sum_{t=1}^{T} F_{M_t}\left(\frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^{b_1} - 1}\right) \tag{52}$$

$$\text{UB}(\mathbb{E}[FR], \theta, b_2) = \frac{1}{T}\sum_{t=1}^{T} F_{M_t}\left(\frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^{b_2} - 1}\right) \tag{53}$$

To compare equation 52 & equation 53, we essentially need to compare $F_{M_t}\left(\frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^{b_1}-1}\right)$ and $F_{M_t}\left(\frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^{b_2}-1}\right)$. By the property of the Cumulative Distribution Function (CDF) $F_{M_t}(x)$, we know that if it is evaluated at $x = x_1$ and $x = x_2$, such that $x_1 < x_2$, then:

$$F_{M_t}(x_1) \leq F_{M_t}(x_2) \qquad \text{when} \quad x_1 < x_2 \tag{54}$$

Hence, to compare $F_{M_t}\left(\frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^{b_1}-1}\right)$ and $F_{M_t}\left(\frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^{b_2}-1}\right)$, we compare $\frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^{b_1}-1}$ and $\frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^{b_2}-1}$.

We know that the model is fixed, hence L_θ, C_θ, and $|\theta|$ remain the same. Furthermore, we also know that $b_1 > b_2$, therefore $\frac{1}{2^{b_1}-1} < \frac{1}{2^{b_2}-1}$. This implies that:

$$\frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^{b_1} - 1} < \frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^{b_2} - 1} \tag{55}$$

Therefore, utilizing the CDF property in equation 54, we obtain:

$$F_{M_t}\left(\frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^{b_1} - 1}\right) \leq F_{M_t}\left(\frac{2L_\theta C_\theta \sqrt{|\theta|}}{2^{b_2} - 1}\right) \tag{56}$$

This further means:

$$\frac{1}{T}\sum_{t=1}^{T} F_{M_t}\left(\frac{2L_\theta C_\theta\sqrt{|\theta|}}{2^{b_1}-1}\right) \leq \frac{1}{T}\sum_{t=1}^{T} F_{M_t}\left(\frac{2L_\theta C_\theta\sqrt{|\theta|}}{2^{b_2}-1}\right) \tag{57}$$

And, finally from equation 52 & equation 53 we obtain the following:

$$\boxed{\text{UB}(\mathbb{E}[FR],\theta,b_1) \leq \text{UB}(\mathbb{E}[FR],\theta,b_2)} \tag{58}$$

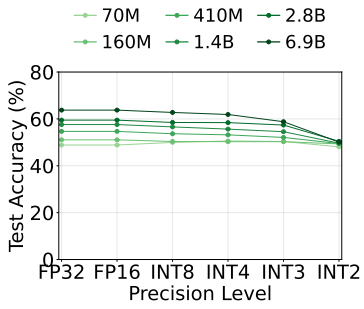
□

E Consistent Test Accuracy Trends under HQQ for Remaining Downstream Datasets

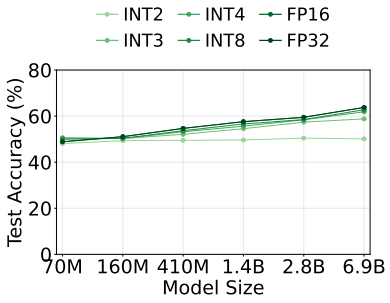
In this section, we provide the test accuracy trends of the impact of PTQ on the remaining downstream datasets: Winogrande, Hellaswag, and PIQA apart from ARC-E, presented across Figs 9a, 9b;Figs. 9c, 9d;Figs 9e, 9f.

E.1 Consistent Trends across Different Post-Training Quantization Techniques

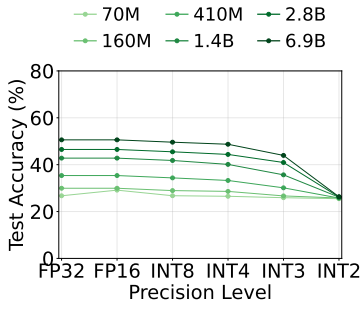
In this section, we verify whether our findings observed in Section 4.1 & 4.2 hold true for all post-training quantization methods—BitsAndBytes, GPTQ, and HQQ. Since BitsAndBytes only supports INT8 and INT4 precisions, we can only provide comparison across INT8 and INT4 precision levels for all PTQ techniques, for Pythia-70M, 160M, 410M, 1.4B and 2.8B models. From Figs. 10 & 11, we can again infer that for a fixed model size, as the model weights' precision level goes down, the verbatim regurgitation rate also reduces (along with the test accuracy) across all PTQ techniques. Apart from precision, we also observe from Fig. 12 that for the same set of sequences, larger quantized models regurgitate more and exhibit higher test accuracy than smaller ones, across all PTQ techniques.



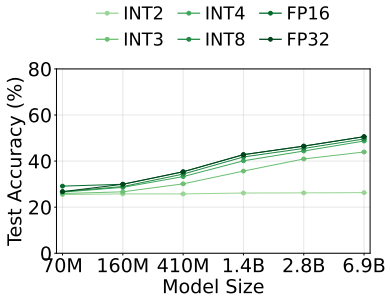
(a) Test Accuracy across Precision Level (Winogrande)



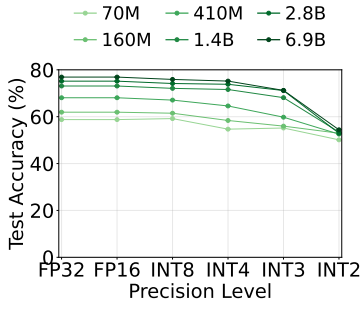
(b) Test Accuracy across Model Size (Winogrande)



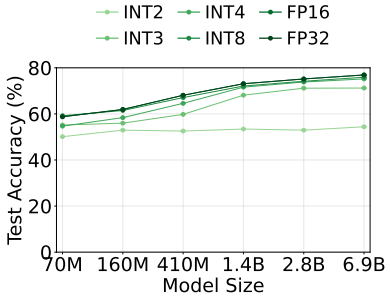
(c) Test Accuracy across Precision Level (Hellaswag)



(d) Test Accuracy across Model Size (Hellaswag)

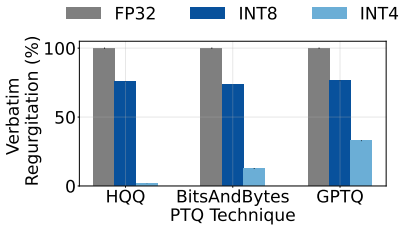


(e) Test Accuracy across Precision Level (PIQA)

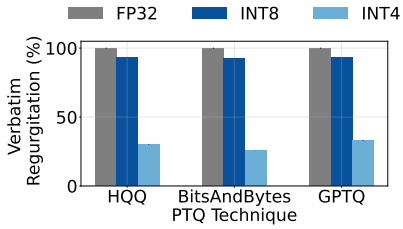


(f) Test Accuracy across Model Size (PIQA)

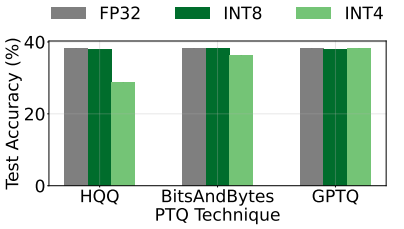
Figure 9: **Consistent Trends on Impact of Quantization on Test Accuracy across Precision Level and Model Size:** Even for other downstream datasets test accuracy decreases as quantization precision decreases, with larger quantized models have better performance than smaller ones (HQQ is used).



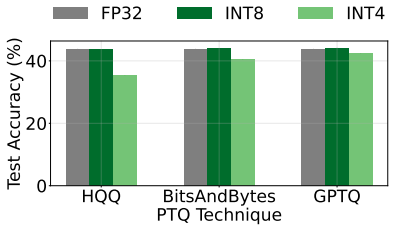
(a) Pythia-70M - Verbatim Regurgitation



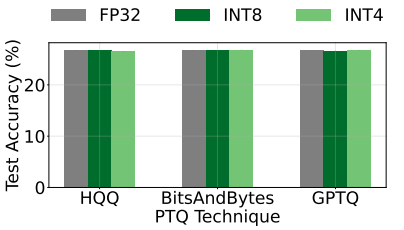
(b) Pythia-160M - Verbatim Regurgitation



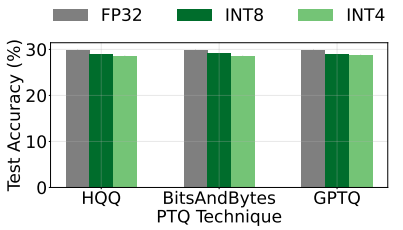
(c) Pythia-70M - Test Accuracy (ARC-E)



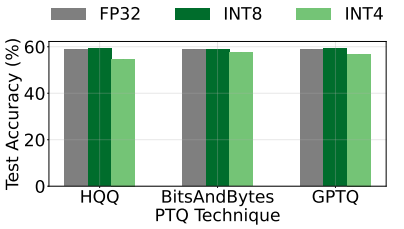
(d) Pythia-160M - Test Accuracy (ARC-E)



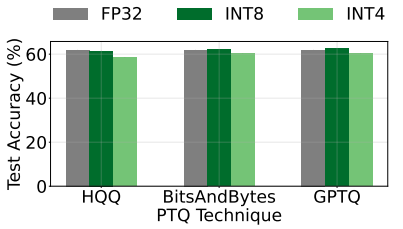
(e) Pythia-70M - Test Accuracy (Hellaswag)



(f) Pythia-160M - Test Accuracy (Hellaswag)

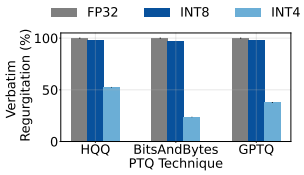


(g) Pythia-70M - Test Accuracy (PIQA)

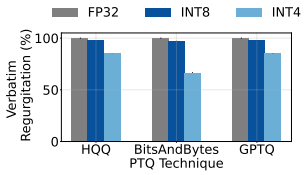


(h) Pythia-160M - Test Accuracy (PIQA)

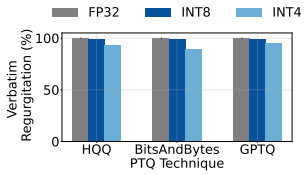
Figure 10: **Consistent trend across all PTQ methods where a drop in precision reduces regurgitation for a fixed model size.** For all PTQ techniques (HQQ, BitsAndBytes, and GPTQ), both verbatim regurgitation rate and test accuracy decrease as the precision level drops during quantization, across all model sizes. Please note that the error bars are invisibly small.



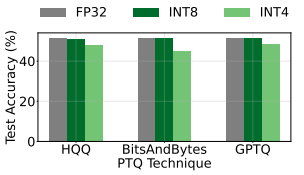
(a) Pythia-410M - Verbatim Regurgitation



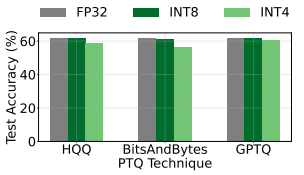
(b) Pythia-1.4B - Verbatim Regurgitation



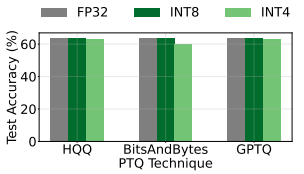
(c) Pythia-2.8B - Verbatim Regurgitation



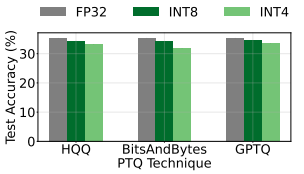
(d) Pythia-410M - Test Accuracy (ARC-E)



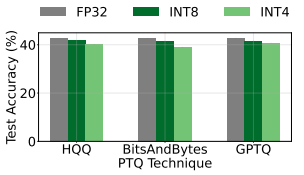
(e) Pythia-1.4B - Test Accuracy (ARC-E)



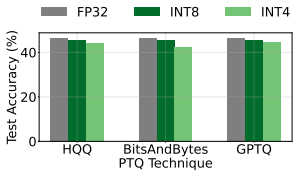
(f) Pythia-2.8B - Test Accuracy (ARC-E)



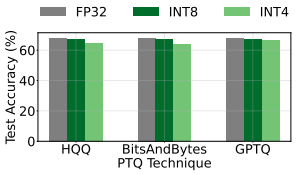
(g) Pythia-410M - Test Accuracy (Hellaswag)



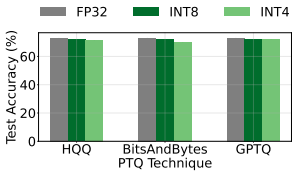
(h) Pythia-1.4B - Test Accuracy (Hellaswag)



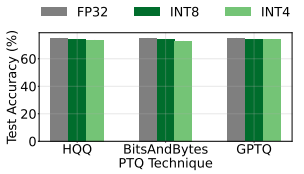
(i) Pythia-2.8B - Test Accuracy (Hellaswag)



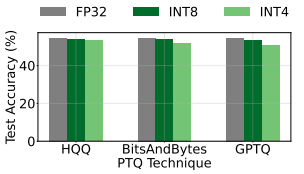
(j) Pythia-410M - Test Accuracy (PIQA)



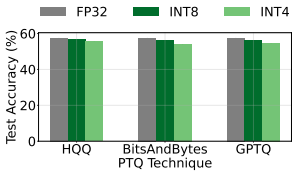
(k) Pythia-1.4B - Test Accuracy (PIQA)



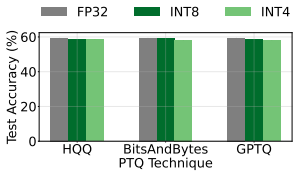
(l) Pythia-2.8B - Test Accuracy (PIQA)



(m) Pythia-410M - Test Accuracy (Winogrande)

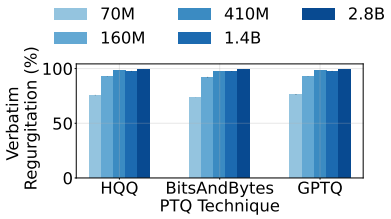


(n) Pythia-1.4B - Test Accuracy (Winogrande)

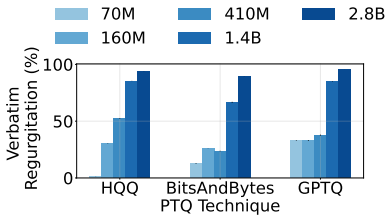


(o) Pythia-2.8B - Test Accuracy (Winogrande)

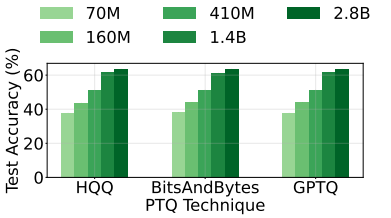
Figure 11: **Consistent trend across all PTQ methods where a drop in precision reduces regurgitation for a fixed model size.** For all PTQ techniques (HQQ, BitsAndBytes, and GPTQ), both verbatim regurgitation rate and test accuracy decrease as the precision level drops during quantization, across all model sizes. Please note that the error bars are invisibly small.



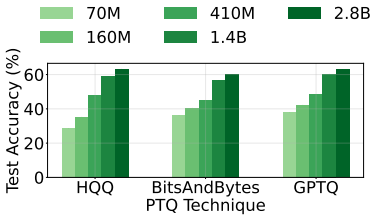
(a) INT8 Precision - Verbatim Regurgitation



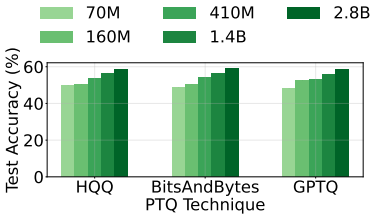
(b) INT4 Precision - Verbatim Regurgitation



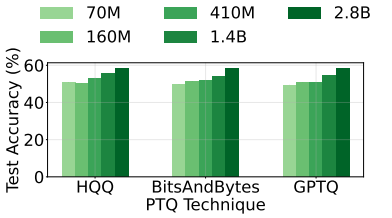
(c) INT8 Precision - Test Accuracy (ARC-E)



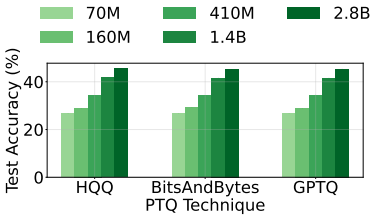
(d) INT4 Precision - Test Accuracy (ARC-E)



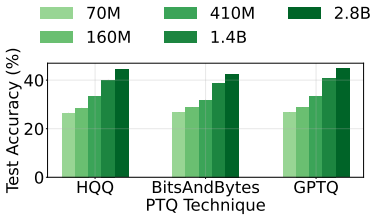
(e) INT8 Precision - Test Accuracy (Winogrande)



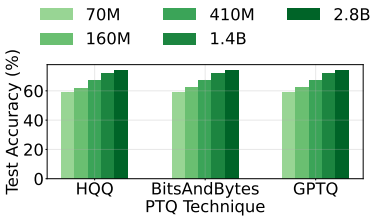
(f) INT4 Precision - Test Accuracy (Winogrande)



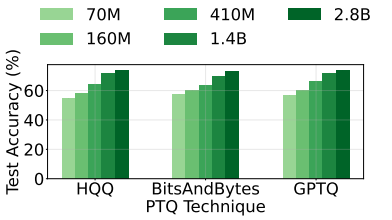
(g) INT8 Precision - Test Accuracy (Hellaswag)



(h) INT4 Precision - Test Accuracy (Hellaswag)



(i) INT8 Precision - Test Accuracy (PIQA)



(j) INT4 Precision - Test Accuracy (PIQA)

Figure 12: **Consistent trend across all PTQ methods where for the same set of sequences, larger quantized models regurgitate more for a fixed precision.** For all PTQ techniques (HQQ, BitsAndBytes, and GPTQ), larger quantized models regurgitate more than smaller models while maintaining higher test accuracy than them across all precision levels, for the same set of sequences. Please note that the error bars are invisibly small.

F Filtration Procedure to study Impact of Quantization on Regurgitation for Varying (Prefix,Suffix) Lengths

To compare the impact of quantization on regurgitation under different (prefix,suffix) lengths, a filtration step is necessary to make sure that we first have a set of sequences which all the 6 models: Pythia-70M, 160M, 410M, 1.4B, 2.8B, and 6.9B, regurgitate in their full precision setting, under all 4 (prefix,suffix) lengths settings: (8,56), (16,48), (24,40), and (32,32). In this manner, we can fairly compare quantization's impact across varying (prefix,suffix) lengths across all models. To come up with this set of sequences, we first filter out sequences which are regurgitated by all 6 models (in FP32 precision) under a fixed (prefix,suffix) length setting, to obtain 4 sets each belonging to a fixed (prefix,suffix) length. We then take an intersection of these 4 sets, to obtain an universal set of sequences which are regurgitated by all 6 models (in FP32 precision) irrespective of the (prefix,suffix) length. The size of this set comes out to be 105,456.

G Scaling Law Analysis

In this section, we provide a thorough analysis on: (1) comparing 4 different scaling law equations including our proposed one for verbatim regurgitation under quantization $\text{VR}(b, n, L)$ and intuition behind our proposed scaling law, and (2) cross-validation evaluation of our scaling law across different model sizes, in Sections G.1 & G.2, respectively.

G.1 Comparison across different Scaling Law Forms

In this study, we discuss three more scaling law forms other than our proposed scaling law, as follows:

1. Power Scaling Law Form: Following prior works in language model scaling dynamics (Kaplan et al., 2020; Hoffmann et al., 2022), power-law formulations serve as the standard benchmark for modeling performance as a function of model and resource scale. We adapt the Chinchilla-style power law to model verbatim regurgitation across quantization precision b, model size N, and prefix length L (please note that we clip the value till 100 because verbatim regurgitation could physically at max reach 100%):

$$\text{VR}(b, N, L) = \min\left(100, c_1 + c_2 \cdot N^{\alpha} + c_3 \cdot b^{\beta} + c_4 \cdot L^{\gamma}\right), \tag{59}$$

where $c_1, c_2, c_3, c_4, \alpha, \beta$, and γ are all parameters to be estimated.

2. Sigmoid Scaling Law Form: As discussed in Section 5, verbatim regurgitation across precision and model size represents a sigmoidal nature. Hence, we adopt the sigmoidal form scaling law to model verbatim regurgitation as:

$$\text{VR}(b, N, L) = \frac{100}{1 + \exp\left(-c_1\left(b - (c_2 - c_3 \log N - c_4 \log L)\right)\right)}, \tag{60}$$

where c_1, c_2, c_3, and c_4 are all parameters to be estimated. Please note that we consider a shifted version of the standard sigmoid function, primarily because we observed that the inflection precision point at which regurgitation begins to decrease varies across model size.

3. Gompertz Scaling Law Form: Apart from the sigmoid scaling curve, we also experiment with an asymmetric sigmoidal curve called Gompertz curve (Gompertz, 1815). We do this primarily because the growth/decay of verbatim regurgitation can be asymmetric before and after the 50% regurgitation mark. Hence, we adapt the Gompertz curve to model verbatim regurgitation as follows:

$$\text{VR}(b, N, L) = 100 \cdot \exp\left(-\exp\left(-c_1\left(b - (c_2 - c_3 \log N - c_4 \log L)\right)\right)\right), \tag{61}$$

where c_1, c_2, c_3, and c_4 are all parameters to be estimated.

4. Our Proposed Scaling Law Form: While both standard Sigmoid and Gompertz formulations effectively capture the general S-shaped decay of verbatim regurgitation under quantization, their reliance on an unbounded shift term ($c_3 \ln N$) fails at extreme scales. Specifically, as model size grows ($N \to \infty$), the $-c_3 \ln N$ term forces the effective bit threshold $b_0(N, L) = c_2 - c_3 \ln N - c_4 \ln L$ to approach negative infinity ($b_0 \to -\infty$). This unphysically implies that a sufficiently large model could achieve 100% verbatim recall at zero or negative bit precision, violating fundamental information-theoretic limits.

To adapt to physical reality, we replace the unbounded linear shift with a *saturating logistic parameterization* over $\log N$. We again use the Gompertz style asymmetric logistic form to model this. Overall, our proposed scaling law is as follows:

$$\text{VR}(b, N, L) = 100 \cdot \exp\left(-\exp\left(-c_1\left(b - \left(c_2 - \frac{c_3}{1 + \exp\left(-c_4(\log N - \log c_5)\right)} - c_6 \log L\right)\right)\right)\right), \tag{62}$$

where c_1, c_2, c_3, c_4, c_5, and c_6 are all parameters to be estimated.

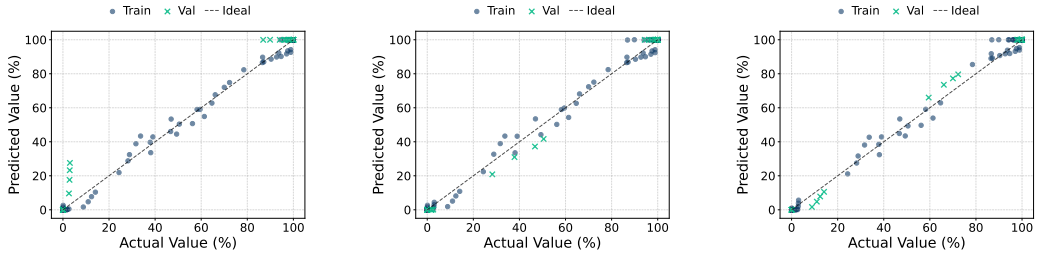
To further validate how these scaling laws perform, we fit them on our observed verbatim regurgitation values across 5 model sizes: 70M, 160M, 410M, 1.4B, 2.8B; 6 precision levels (in bits): 32, 16, 8, 4, 3, 2; and prefix lengths: 8, 16, 24, 32, and evaluate on the held-out largest model 6.9B across all 6 precision levels and 4 prefix lengths. We evaluate these scaling laws by two metrics: R^2-score and Mean Absolute Error (MAE (%)) for train and validation sets. The results are presented as follows:

Scaling Law Form	**Train R^2-score**	**Train MAE (%)**	**Val R^2-score**	**Val MAE (%)**
Power Law	0.971	3.802	0.594	14.139
Sigmoid	0.993	1.950	0.886	7.407
Gompertz	0.993	1.785	0.910	6.125
Our Scaling Law	**0.995**	**1.610**	**0.988**	**2.201**

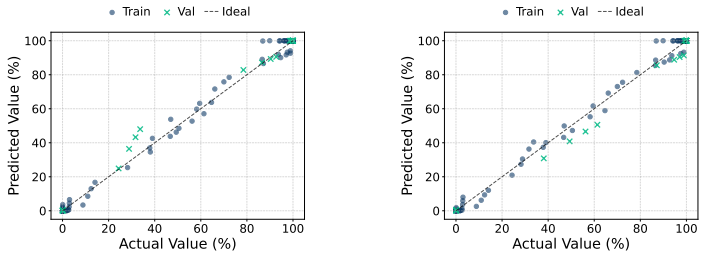
Table 1: Cross-validation performance (R^2 and MAE) of scaling laws fitted on 70M–2.8B models and evaluated on the held-out 6.9B model across 6 precision levels and 4 prefix lengths. The best performing scaling law is **bolded**. From the results, we can clearly observe that our proposed scaling law performs much better than existing ones in modeling verbatim regurgitation under quantization precision, model size and prefix-length.

G.2 Cross-Validation Evaluation of the Proposed Scaling Law across different Model Sizes

In this section, we examine how our scaling law performs when we do a cross-validation performance across other model sizes apart from 6.9B. The results are presented across Figs 6, 13 and Table 2.



(a) Held-out: 70M (b) Held-out: 160M (c) Held-out: 410M



(d) Held-out: 1.4B (e) Held-out: 2.8B

Figure 13: Predicted vs. actual verbatim regurgitation percentages (%) under Leave-One-Out (LOO) cross-validation across remaining five model sizes for the proposed Scaling Law.

Held-Out Model	Train R^2	Train MAE (%)	Val R^2	Val MAE (%)
70M	0.996	1.408	0.970	4.531
160M	0.995	1.601	0.992	2.339
410M	0.995	1.511	0.992	2.335
1.4B	0.996	1.581	0.988	1.922
2.8B	0.996	1.502	0.986	2.527
6.9B	0.995	1.610	0.988	2.201

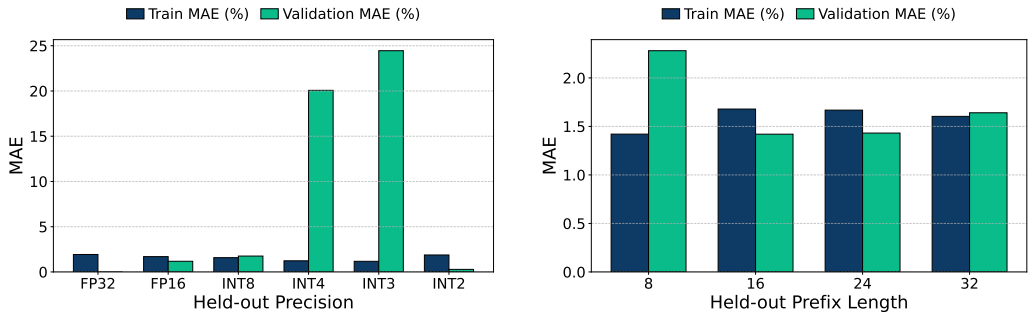
Table 2: Train and validation performance (R^2 score and MAE) of the proposed scaling law across held-out model folds.

From the results, we consistently observe that for most of the model sizes: 160M, 410M, 1.4B, 2.8B and 6.9B, the validation MAE scores are well within 1-2.5%, except for 70M model size, where we observe a much higher validation MAE of 4.531%.

G.3 Cross-Validation Evaluation of the Proposed Scaling Law across different Precision Levels and Prefix Lengths

In this section, we examine how our scaling law performs when we do a cross-validation performance across different precision levels (Fig. 14a) and prefix length (Fig. 14b).

Precision levels. Fig. 14a shows a sharp asymmetry across the six precision levels. When FP32, FP16, INT8, or INT2 is held out, validation MAE remains small (0.05–2%). When INT4 or INT3 is held out, validation MAE jumps to 20–25%. This gap is not a failure of the scaling law's functional



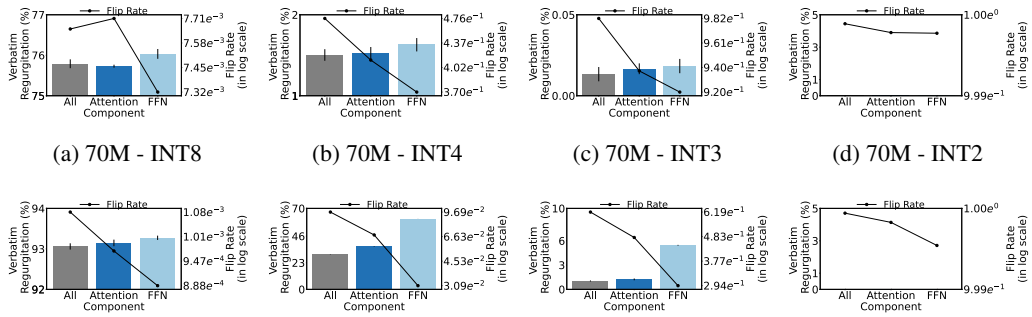
(a) Cross-validation across precision levels (b) Cross-validation across prefix lengths

Figure 14: Leave-one-out cross-validation analysis of the proposed scaling law across precision levels (left) and prefix lengths (right).

form but rather a consequence of where each precision level sits on the underlying sigmoidal curve. FP32, FP16, and INT8 sit on the curve's saturated upper tail (VR $\approx 100\%$) and INT2 sits on its saturated lower tail (VR $\approx 0\%$) across essentially all model sizes (see Fig. 1). Hence, removing any one of these leaves the remaining five precisions sufficient to predict the missing tail precisions values. INT3 and INT4, in contrast, lie inside the transition region of the curve, where VR changes most steeply with bit-width and varies the most with model scale (see Fig. 1). Therefore, removing either one leaves the fit with only tail points and a single transition point (INT4/INT3 depending on which precision was left out) to work with—not enough to reconstruct the curve's steep middle section. That is why the law is challenging to fit when either INT4 or INT3 is omitted.

Prefix lengths. Fig. 14b shows a much more uniform picture: validation MAE stays within a narrow 1–2.5% band across all four held-out prefix lengths, with prefix length 8 showing a marginally higher error than the rest.

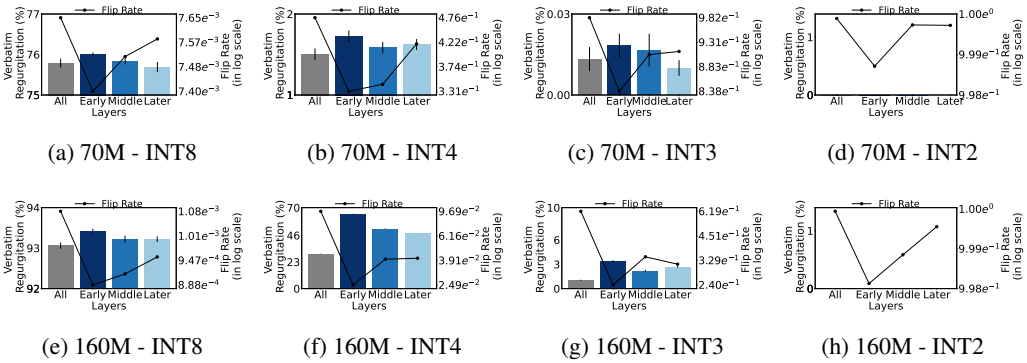
H Consistent Trends for Impact of Component-Wise Quantization on Verbatim Regurgitation



(a) 70M - INT8 (b) 70M - INT4 (c) 70M - INT3 (d) 70M - INT2

(e) 160M - INT8 (f) 160M - INT4 (g) 160M - INT3 (h) 160M - INT2

Figure 15: **Quantizing Attention module has a larger impact on verbatim regurgitation than FFN.** Quantizing attention modules leads to a higher reduction in regurgitation compared to FFN, correlating with the higher flip rate (depicted by the black line) corresponding to attention modules quantization than the FFN modules (HQQ is used).



(a) 70M - INT8 (b) 70M - INT4 (c) 70M - INT3 (d) 70M - INT2

(e) 160M - INT8 (f) 160M - INT4 (g) 160M - INT3 (h) 160M - INT2

Figure 16: **Later layers quantization has a larger impact on verbatim regurgitation.** Quantizing later layers leads to a higher reduction in regurgitation compared to early and middle layers. This trend correlates further with the higher flip rate for later layers quantization (depicted by the black line) in comparison to early-middle layers (HQQ is used).

I Consistent Trends for Impact of Layer-Wise Quantization on Verbatim Regurgitation

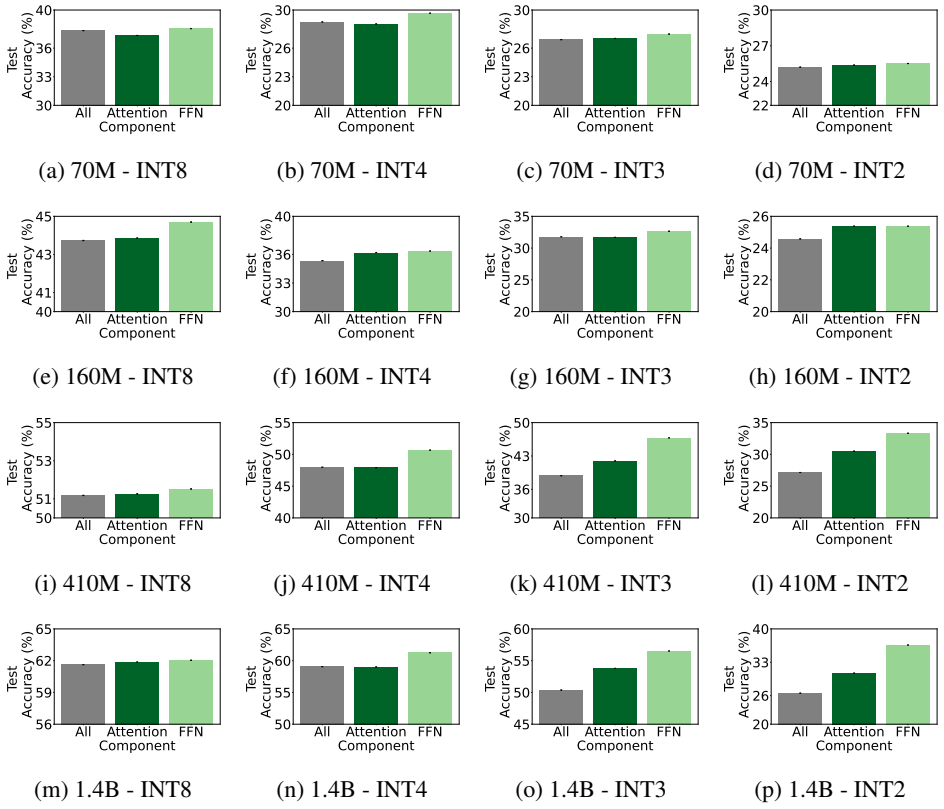
J Impact of Later Layers and Attention Modules Quantization is Less Significant for Certain Instances

We note that the observed phenomena of higher impact of later layers (neither early nor middle) and attention modules (not FFN) on regurgitation are less pronounced at the extremes of precision (INT8 and INT2). At higher precision (INT8), the overall impact on regurgitation is limited, resulting in smaller gaps between early, middle, and late layers, as well as between attention and FFN components. At very low precision (INT2), regurgitation rates approach zero for smaller models (Pythia-70M and 160M), which similarly reduces observable differences. These small gaps are consistent with the broader trends observed in Sections 4.1 & 4.2, where higher precision levels (INT8) have very less impact on regurgitation and low precision levels (INT2) drive regurgitation to near zero. Despite these boundary effects, the overall patterns remain consistent: (i) quantizing later layers has a larger impact on reducing regurgitation than quantizing early-middle layers, and (ii) quantizing attention modules has a higher impact on regurgitation than quantizing FFN components.

K Impact of Component-wise Quantization on Test Accuracy

In this section, we investigate the impact of quantizing Attention vs FFN modules across all 4 downstream tasks: ARC-E, Hellaswag, PIQA, and Winogrande, across Figs. 17, 18, 19, and 20. We observe that for the ARC-E dataset, quantizing attention modules has a relatively higher impact on test accuracy than quantizing FFN modules across model sizes and precisions. However, for all the remaining 3 datasets–Hellaswag, PIQA, and Winogrande–there is no consistent significant impact on test accuracy when compared among attention vs FFN modules quantization.

K.1 ARC-E dataset



(a) 70M - INT8 (b) 70M - INT4 (c) 70M - INT3 (d) 70M - INT2

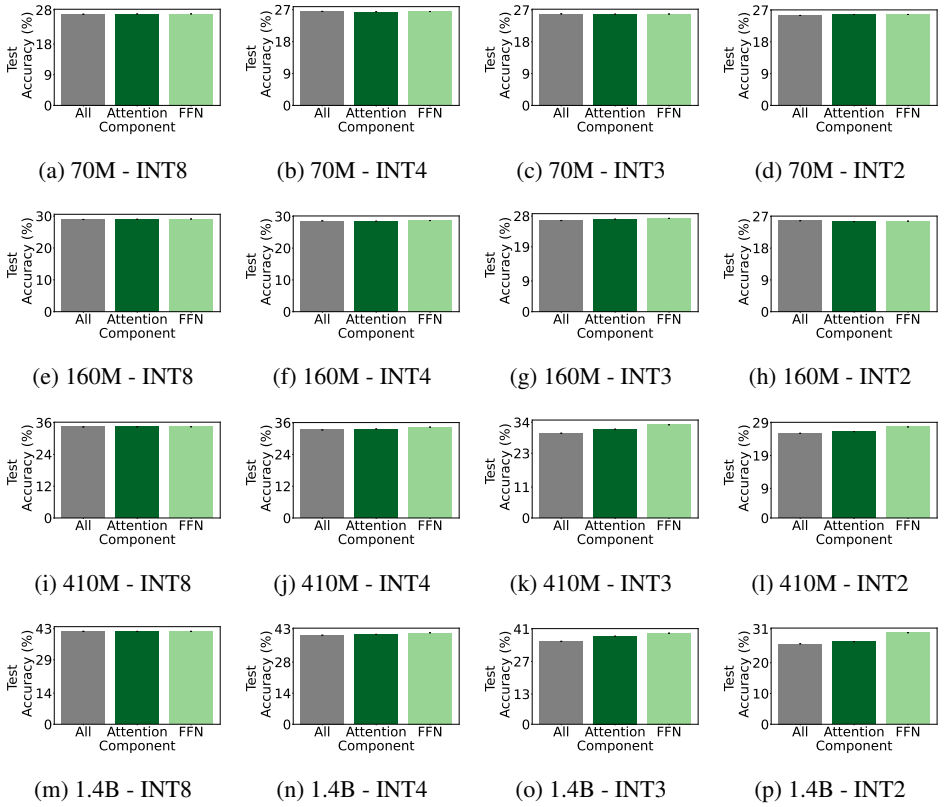
(e) 160M - INT8 (f) 160M - INT4 (g) 160M - INT3 (h) 160M - INT2

(i) 410M - INT8 (j) 410M - INT4 (k) 410M - INT3 (l) 410M - INT2

(m) 1.4B - INT8 (n) 1.4B - INT4 (o) 1.4B - INT3 (p) 1.4B - INT2

Figure 17: Quantizing the attention modules lead to a relatively higher drop in test accuracy compared to FFN modules quantization for ARC-E. Please note that the error bars are invisibly small (HQQ is used).

K.2 HELLASWAG DATASET



(a) 70M - INT8 (b) 70M - INT4 (c) 70M - INT3 (d) 70M - INT2

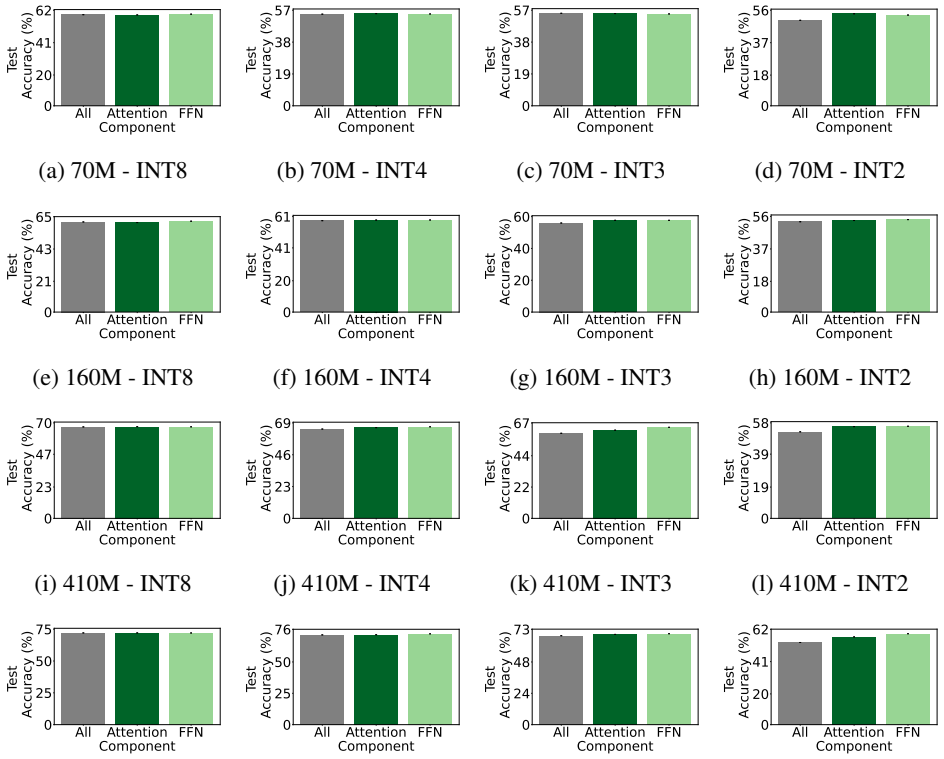
(e) 160M - INT8 (f) 160M - INT4 (g) 160M - INT3 (h) 160M - INT2

(i) 410M - INT8 (j) 410M - INT4 (k) 410M - INT3 (l) 410M - INT2

(m) 1.4B - INT8 (n) 1.4B - INT4 (o) 1.4B - INT3 (p) 1.4B - INT2

Figure 18: Quantizing attention and FFN modules separately results in a broadly aligning impact on test accuracy across model sizes and precision levels for Hellaswag. Please note that the error bars are invisibly small (HQQ is used).

K.3 PIQA DATASET



(a) 70M - INT8 (b) 70M - INT4 (c) 70M - INT3 (d) 70M - INT2

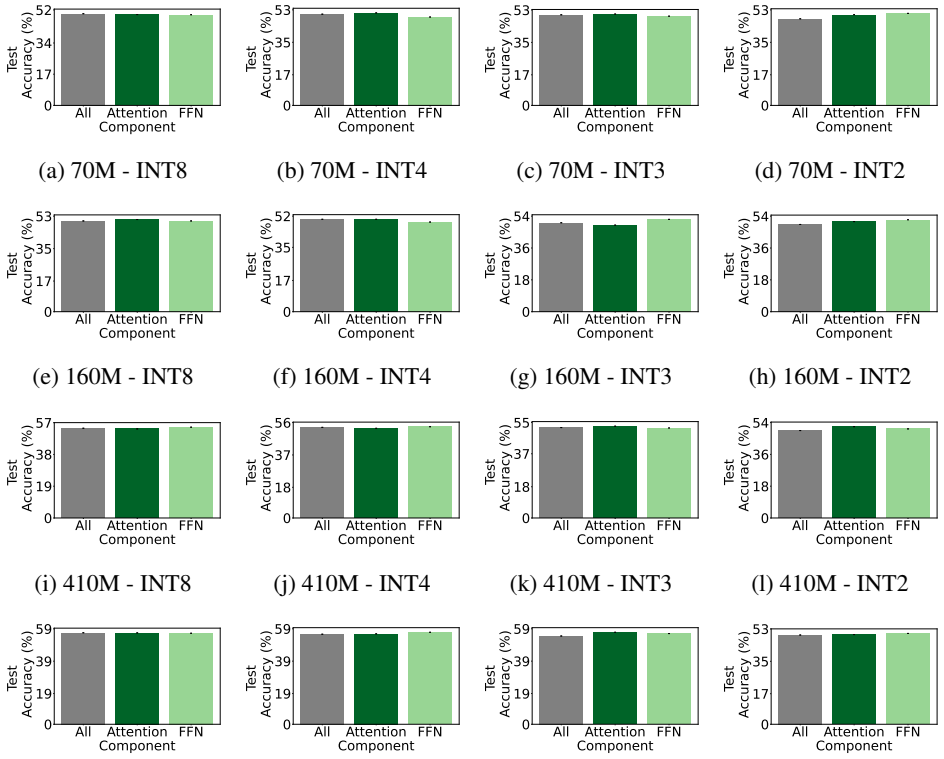
(e) 160M - INT8 (f) 160M - INT4 (g) 160M - INT3 (h) 160M - INT2

(i) 410M - INT8 (j) 410M - INT4 (k) 410M - INT3 (l) 410M - INT2

(m) 1.4B - INT8 (n) 1.4B - INT4 (o) 1.4B - INT3 (p) 1.4B - INT2

Figure 19: Quantizing attention and FFN modules separately results in a broadly aligning impact on test accuracy across model sizes and precision levels for PIQA. Please note that the error bars are invisibly small (HQQ is used).

K.4 WINOGRANDE DATASET



(a) 70M - INT8 (b) 70M - INT4 (c) 70M - INT3 (d) 70M - INT2

(e) 160M - INT8 (f) 160M - INT4 (g) 160M - INT3 (h) 160M - INT2

(i) 410M - INT8 (j) 410M - INT4 (k) 410M - INT3 (l) 410M - INT2

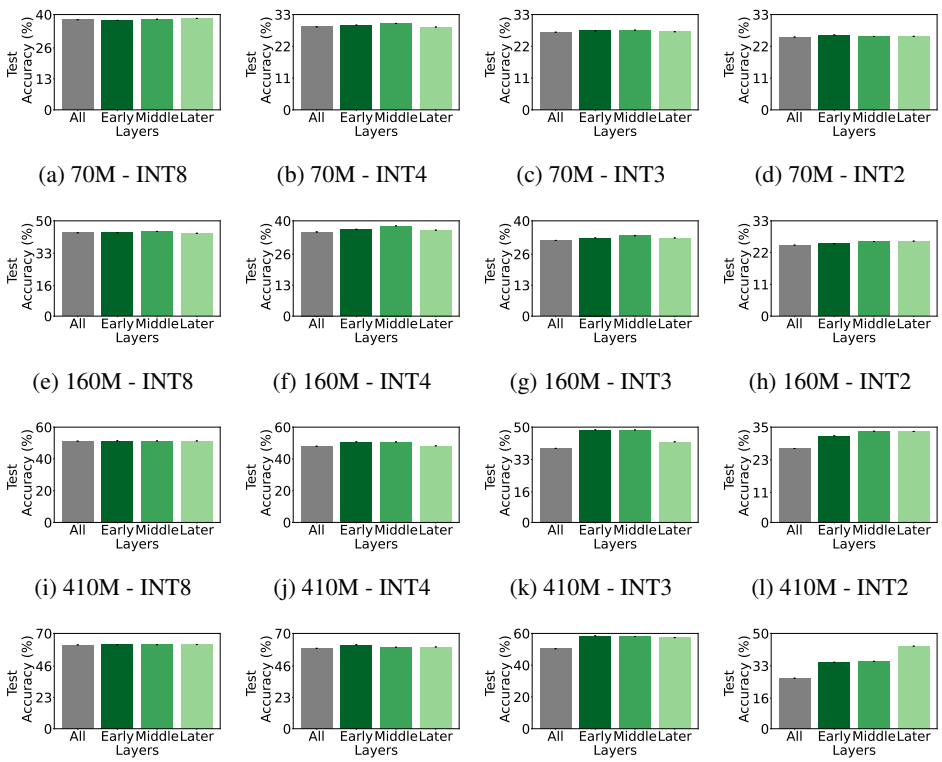
(m) 1.4B - INT8 (n) 1.4B - INT4 (o) 1.4B - INT3 (p) 1.4B - INT2

Figure 20: Quantizing attention and FFN modules separately results in a broadly similar impact on test accuracy across model sizes and precision levels for Winogrande. Please note that the error bars are invisibly small (HQQ is used).

L IMPACT OF LAYER-WISE QUANTIZATION ON TEST ACCURACY

In this section, we discuss the impact of quantizing early vs. middle vs. late layers for all 4 downstream tasks: ARC-E, Hellaswag, PIQA, and Winogrande, across Figs. 21, 22, 23, and 24. We observe that for the 4 datasets–ARC-E, Hellaswag, PIQA, and Winogrande–there is no consistent significant impact on test accuracy when compared among early, middle, and late layers quantization.

L.1 ARC-E DATASET



(a) 70M - INT8 (b) 70M - INT4 (c) 70M - INT3 (d) 70M - INT2

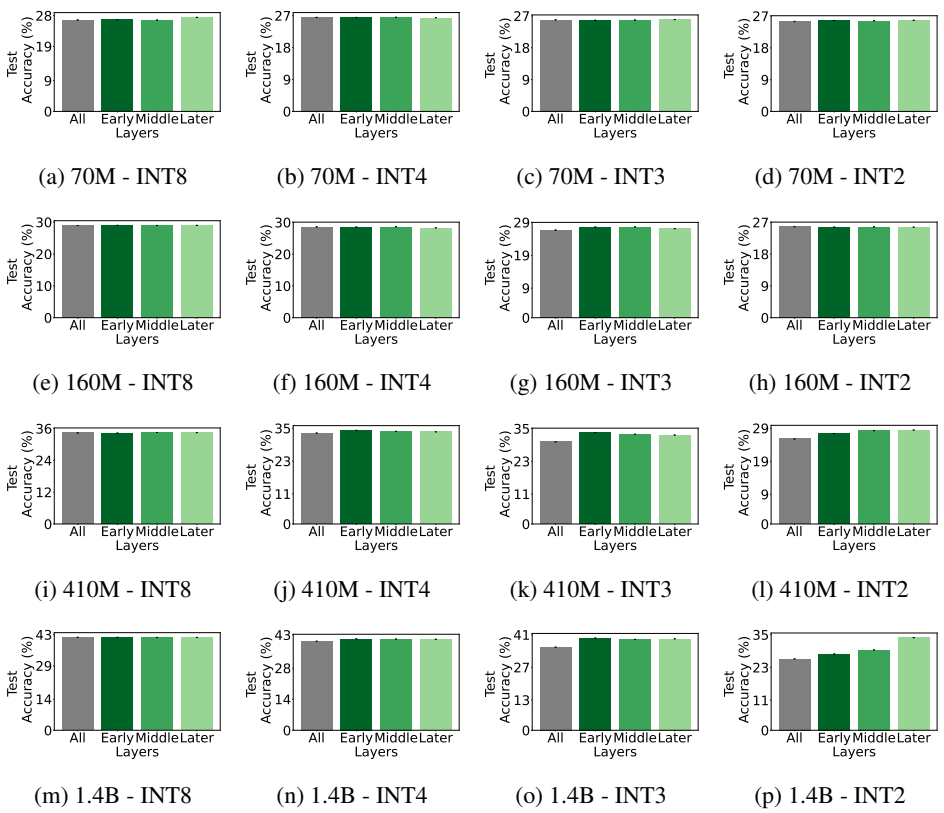
(e) 160M - INT8 (f) 160M - INT4 (g) 160M - INT3 (h) 160M - INT2

(i) 410M - INT8 (j) 410M - INT4 (k) 410M - INT3 (l) 410M - INT2

(m) 1.4B - INT8 (n) 1.4B - INT4 (o) 1.4B - INT3 (p) 1.4B - INT2

Figure 21: Early, middle and later layers quantization have no significant impact on test accuracy when compared among themselves for the ARC-E dataset. Please note that the error bars are invisibly small (HQQ is used).

L.2 HELLASWAG DATASET



(a) 70M - INT8 (b) 70M - INT4 (c) 70M - INT3 (d) 70M - INT2

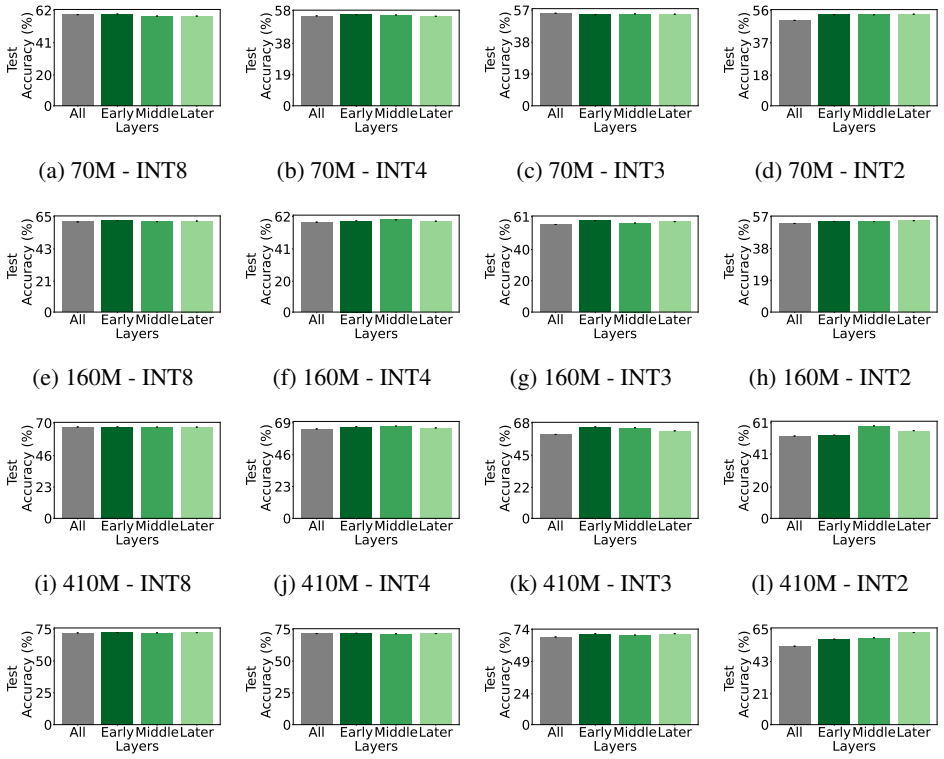
(e) 160M - INT8 (f) 160M - INT4 (g) 160M - INT3 (h) 160M - INT2

(i) 410M - INT8 (j) 410M - INT4 (k) 410M - INT3 (l) 410M - INT2

(m) 1.4B - INT8 (n) 1.4B - INT4 (o) 1.4B - INT3 (p) 1.4B - INT2

Figure 22: Early, middle and later layers quantization have no significant impact on test accuracy when compared among themselves for the Hellaswag dataset. Please note that the error bars are invisibly small (HQQ is used).

L.3 PIQA DATASET



(a) 70M - INT8 (b) 70M - INT4 (c) 70M - INT3 (d) 70M - INT2

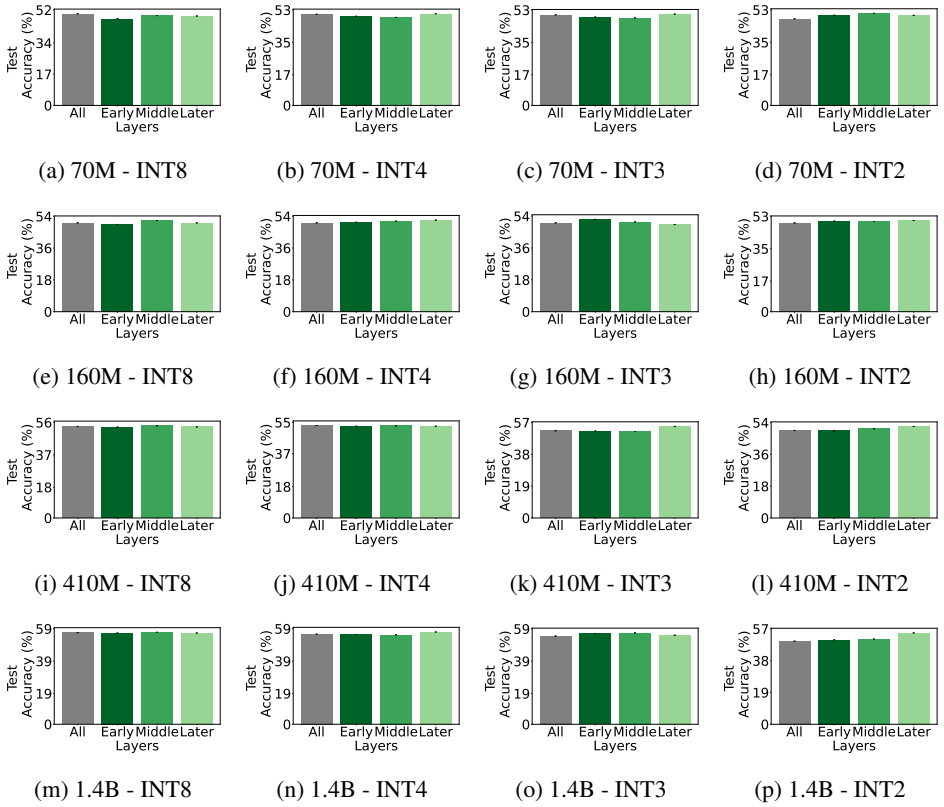
(e) 160M - INT8 (f) 160M - INT4 (g) 160M - INT3 (h) 160M - INT2

(i) 410M - INT8 (j) 410M - INT4 (k) 410M - INT3 (l) 410M - INT2

(m) 1.4B - INT8 (n) 1.4B - INT4 (o) 1.4B - INT3 (p) 1.4B - INT2

Figure 23: Early, middle and later layers quantization have no significant impact on test accuracy when compared among themselves for the PIQA dataset. Please note that the error bars are invisibly small (HQQ is used).

L.4 WINOGRANDE DATASET



(a) 70M - INT8 (b) 70M - INT4 (c) 70M - INT3 (d) 70M - INT2

(e) 160M - INT8 (f) 160M - INT4 (g) 160M - INT3 (h) 160M - INT2

(i) 410M - INT8 (j) 410M - INT4 (k) 410M - INT3 (l) 410M - INT2

(m) 1.4B - INT8 (n) 1.4B - INT4 (o) 1.4B - INT3 (p) 1.4B - INT2

Figure 24: Early, middle and later layers quantization have no significant impact on test accuracy when compared among themselves for the Winogrande dataset. Please note that the error bars are invisibly small (HQQ is used).

M LIMITATIONS

Our scope is to investigate the impact of post-training quantization (PTQ) on verbatim regurgitation by utilizing the Pythia model suite, which is the only open source suite with regurgitated sequences across multiple model sizes. Upon future availability of other models, extending this analysis to other model families and quantization-aware training (QAT) would be a natural and interesting future direction.