ConQuR: Corner Aligned Activation Quantization via Optimized Rotations for LLMs

Chayne Thrash
Department of Computer Science
Vanderbilt University
chayne.thrash@vanderbilt.edu

Ali Abbasi
Department of Computer Science
Vanderbilt University
ali.abbasi@vanderbilt.edu

Soheil Kolouri
Department of Computer Science
Vanderbilt University
soheil.kolouri@vanderbilt.edu

Abstract

Large language models (LLMs) are costly to deploy due to their large memory footprint and high inference cost. Weight-activation quantization can reduce these costs, but low-bit activation quantization remains difficult because activation outliers induce large quantization error. Recent rotation-based methods address this by applying orthogonal transformations that redistribute activation magnitude across dimensions, but existing approaches either require expensive end-to-end rotation training or rely on stored activation corpora, introducing significant compute or storage overhead. We propose a lightweight post-training rotation calibration method for LLM activation quantization. Our method learns orthogonal rotations that align normalized activations with the corners of an inscribed hypercube, encouraging activation energy to be distributed more evenly across dimensions. This objective admits an efficient closed-form update via the orthogonal Procrustes problem, avoiding gradient-based optimization over the orthogonal group. We further introduce an online calibration procedure that updates rotations as calibration samples are processed, eliminating the need to store activations on disk and allowing rotations to adapt to quantized activation distributions during calibration. Experiments on Llama-2 and Llama-3 models from 3B to 70B parameters show that our method achieves competitive or improved performance across perplexity benchmarks and common sense reasoning tasks while avoiding both costly end-to-end training and large offline activation storage. Our code is publicly available at this URL.

1 Introduction

Large language models (LLMs) [1, 2, 3] have achieved remarkable performance across a wide range of domains, including natural language understanding and generation, multi-step reasoning, code synthesis, and mathematical problem solving [3, 4, 5, 6]. However, these capabilities are enabled by models with billions of parameters, making inference expensive in terms of memory footprint, latency, and energy consumption. These costs limit the deployment of LLMs in practical settings, particularly on resource-constrained hardware or in applications requiring high-throughput inference.

A variety of compression techniques have been developed to reduce the cost of deploying large models. These include pruning methods, which remove unimportant weights or structures from the model [7, 8, 9]; singular value decomposition (SVD) and low-rank approximation methods, which replace dense weight matrices with compact low-rank factors [10, 11]; and knowledge distillation,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

which transfers the behavior of a large teacher model to a smaller student model [12]. Among these approaches, quantization has emerged as one of the most effective and widely used strategies for improving inference efficiency.

Quantization reduces the memory and compute requirements of LLMs by representing weights, activations, or KV-cache tensors using lower-precision numerical formats. Weight-only quantization methods have shown that LLM weights can often be compressed to very low bit-widths while preserving model quality [13, 14]. However, quantizing activations remains substantially more challenging. A primary obstacle is the presence of large activation outliers, which expand the dynamic range of the quantizer and reduce the effective resolution assigned to the majority of activation values [15, 16]. As a result, low-bit weight-activation quantization typically requires additional mechanisms to control or suppress activation outliers.

Several approaches have been proposed to address this challenge. Mixed-precision methods allocate higher precision to sensitive layers, channels, or activation groups [17, 18]. Smoothing-based methods use diagonal rescaling transformations to migrate activation outliers into the weights, thereby reducing the activation quantization range [15, 19]. More recently, rotation-based methods have emerged as a promising alternative for mitigating activation outliers. These methods apply orthogonal transformations to hidden states, spreading outlier dimensions across the feature space while preserving the model function by merging inverse rotations into adjacent linear layers [20, 21, 22, 23]. Unlike diagonal smoothing, rotations can redistribute activation energy across all dimensions, making them particularly attractive for low-bit activation quantization.

Existing rotation-based quantization methods typically follow one of two strategies: they either optimize rotations end-to-end using a task loss, which can require substantial compute, or they calibrate rotations using objectives that explicitly reshape activation distributions. While the latter avoids the cost of end-to-end fine-tuning, it often requires storing large collections of intermediate activations, leading to substantial storage overhead. Moreover, because these activations are collected before optimization, the calibration procedure is decoupled from the quantized forward pass that will be used at inference time.

In this work, we address these limitations with a rotation-based method for improving LLM activation quantization by aligning normalized activations with vertices of an inscribed hypercube. This objective encourages each activation vector to distribute its magnitude more evenly across feature dimensions, reducing the dominance of outlier coordinates that otherwise determine the quantization scale. Building on the observation that rotation calibration can be cast as an orthogonal Procrustes problem, our method optimizes this geometric objective with lightweight closed-form updates. The resulting online mini-batch procedure calibrates rotations from a small number of full sequences at a time, without storing a large activation corpus, and can update rotations using the quantized activations produced by the current model. This matches calibration more closely to inference-time conditions while retaining the efficiency of post-training rotation calibration.

2 Related work

Post-training quantization and activation outliers. Post-training quantization (PTQ) reduces the memory and inference cost of LLMs without expensive retraining. Weight-only methods such as GPTQ/OPTQ [13], AWQ [14], and SqueezeLLM [24] achieve strong compression by minimizing layer-wise reconstruction error or protecting activation-sensitive weights, while systems such as ZeroQuant combine weight and activation quantization in a full PTQ pipeline [17]. However, weight-only quantization leaves activations in floating point and therefore does not fully realize the efficiency benefits of low-precision inference. Low-bit weight-activation quantization is more challenging because LLM activations contain large, structured outliers that expand the quantization range and reduce effective resolution. Prior work addresses this with equivalent transformations: SmoothQuant migrates quantization difficulty from activations into weights using per-channel scaling [15], Outlier Suppression+ combines shifting and scaling to reduce asymmetric activation outliers [19], and OmniQuant learns quantization parameters and equivalent transformations in a block-wise PTQ framework [18].

Rotation-based quantization and incoherence processing. A closely related family of methods uses orthogonal transformations to distribute outliers across dimensions while preserving model

outputs in full precision. QuIP introduced incoherence processing for LLM quantization, using randomized orthogonal transformations to make weight matrices more amenable to low-bit quantization and providing theoretical guarantees for 2-bit PTQ [25]. QuIP# and QTIP then extended this idea to low-bit vector quantization [26, 27]. Although these methods primarily target weight-only quantization, they established the utility of orthogonal transformations for reshaping tensor distributions before quantization.

Recent work has adapted this idea to activation quantization. QuaRot applies fixed Hadamard rotations throughout the Transformer to remove activation outliers, enabling 4-bit quantization of weights, activations, and KV caches with efficient inference kernels [20]. DuQuant combines rotations and permutations to distribute both normal and massive activation outliers across channels and blocks [28]. SpinQuant shows that the choice of rotation can significantly affect quantized accuracy and learns rotation matrices directly to improve downstream quantized performance [21]. DartQuant further proposes distribution-aware rotational calibration, reducing the computational and memory burden of learning rotations while improving scalability to larger models [22]. Most similar to our work, DFRot avoids the complexity of gradient descent on the orthogonal group by instead directly optimizing quantization error via closed form Procrustes updates [23].

Our work builds on this line of rotation-based quantization methods. DartQuant [22] calibrates rotations using stored activation distributions, while DFRot [23] uses Procrustes-based optimization to directly reduce activation quantization error. Rather than optimizing quantization error on a small fixed activation subset, we optimize a hypercube-corner alignment objective and apply the resulting Procrustes updates online over successive mini-batches of full sequences. This avoids large activation storage and allows rotations to adapt under active activation quantization to the distributions encountered during quantized inference.

3 Method

In this section, we present a rotation-based method for improving activation quantization by aligning activations with the corners of a hypercube. Figure 1 provides an overview of the approach. An overview of how rotations are applied within the LLM can be seen in panel (c) and a full description is provided in Appendix A. We begin by introducing our optimization objective and relate it directly to quantization error. Then, we show that this objective admits an orthogonal Procrustes solution. Finally, we describe how we apply this optimization within an online calibration algorithm.

3.1 Aligning activations with corners of a hyper-cube

A central challenge in activation quantization for large language models is that the hidden state often uses coordinates unevenly: a small subset of dimensions may dominate the magnitude of the activation, while many others remain comparatively small. Such imbalance leads to inefficient use of the quantization grid, since a few large coordinates determine the scale while the remaining coordinates are quantized with unnecessarily coarse resolution. We address this issue by learning an orthogonal rotation that makes normalized activations more balanced across dimensions.

Let $x^i \in \mathbb{R}^d$ denote an activation vector collected from the model, where i indexes activations across a calibration set. We optimize an orthogonal matrix $R \in O(d)$ by solving

$$\min_{R \in O(d)} \sum_i \left\| R \frac{x^i}{\|x^i\|} - z^i \right\|_2^2 \tag{1}$$

where the target $z^i \in \mathbb{R}^d$ is defined coordinatewise as

$$z^i_j = \frac{1}{\sqrt{d}} \text{sign}((Rx^i)_j). \tag{2}$$

The target z^i is a vertex of the hypercube inscribed in the unit hypersphere. Indeed, each coordinate of z^i has magnitude $1/\sqrt{d}$, so $\left\|z^i\right\|_2 = 1$. Geometrically, the objective encourages the rotated activations to move towards a direction in which all coordinates have equal absolute magnitude.

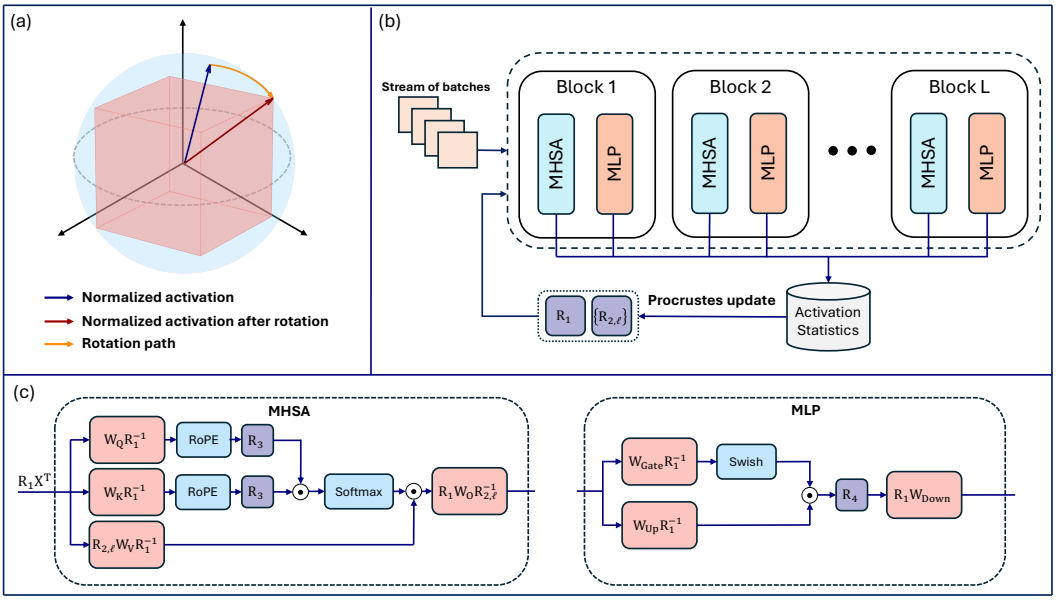

Figure 1: Overview of the proposed rotation-based calibration method. (a) Our method learns an orthogonal rotation that aligns normalized activation vectors with vertices of an inscribed hypercube, encouraging activation magnitude to be distributed more evenly. (b) During calibration, activations are processed online in mini-batches with closed-form orthogonal Procrustes updates. (c) At inference, learned rotations $R_1, \{R_{2,\ell}\}_{l=1}^{L}$are folded into linear layer weights and Hadamard rotations, R_3 and R_4, are applied online.

As no single dimension is allowed to dominate the representation, this optimization encourages an activation's features to be distributed more evenly across its dimensions.

3.2 From quantization error to the corner objective

For a uniform activation quantizer whose scale is determined by the per-token dynamic range, the worst-case squared error is controlled by the largest coordinate magnitude of the activation. For a normalized activation $\tilde{x}$ and an orthogonal rotation R, the ℓ_2 norm is fixed, so rotation can only improve this bound by reducing $\|R\tilde{x}\|_\infty$. The smallest possible value of $\|y\|_\infty$ on the unit sphere is $1/\sqrt{d}$, achieved exactly when all coordinates have equal magnitude, i.e., at the vertices of the hypercube inscribed in the unit sphere.

Directly minimizing $\|R\tilde{x}\|_\infty$ is non-smooth and depends only on the largest coordinate. We instead use a dense dual surrogate. Since ℓ_1 and ℓ_∞ are dual norms under Hölder's inequality, and since $\|y\|_2 = 1$ is fixed, maximizing $\|y\|_1$ encourages the opposite geometry of large $\|y\|_\infty$: it rewards spreading mass evenly across coordinates. In particular, Cauchy–Schwarz gives

$$\|y\|_1 \le \sqrt{d}\|y\|_2 = \sqrt{d},$$

with equality if and only if $|y_1| = \cdots = |y_d| = 1/\sqrt{d}$. Thus, on the unit sphere, maximizing $\|R\tilde{x}\|_1$ and minimizing $\|R\tilde{x}\|_\infty$ have the same per-sample optima: the hypercube corners.

Our corner-distance objective is exactly this ℓ_1 surrogate in squared-distance form. For $z^i = d^{-1/2}\,\mathrm{sign}(R\tilde{x}^i)$,

$$\sum_i \|R\tilde{x}^i - z^i\|_2^2 = 2n - \frac{2}{\sqrt{d}}\sum_i \|R\tilde{x}^i\|_1$$

where n is the number of activations. Therefore, minimizing distance to the nearest hypercube corner is equivalent to maximizing the average ℓ_1 norm of the rotated normalized activations. This gives a dense objective with the same ideal geometry as the ℓ_∞ dynamic-range objective, while leading to the closed-form Procrustes update described below. Appendix B provides the full derivation from symmetric quantization error to the ℓ_∞ dynamic-range ratio, as well as the extension to asymmetric zeropoint quantizers.

Algorithm 1 Online calibration of shared and layerwise rotations

Require: calibration sequences $\mathcal{D} = \{s_i\}_{i=1}^{N}$, mini-batch size B, number of transformer blocks L, initial shared rotation $R_1 \in \mathbb{R}^{d\times d}$, initial block-diagonal rotations $\{R_{2,\ell}\}_{\ell=1}^{L}$ with $R_{2,\ell} \in \mathbb{R}^{n_h d_h \times n_h d_h}$

for mini-batch $\mathcal{B} \subset \mathcal{D}$ of size B **do**
  Initialize empty batch activations $\mathcal{X}_1$ associated with R_1
  Run the model on $\mathcal{B}$ layer by layer
  for $\ell = 1, \ldots, L$ **do**
    Collect self-attention and MLP input activations $X_{\text{attn}}^{(\ell)}$, $X_{\text{mlp}}^{(\ell)}$
    Accumulate $X_{\text{attn}}^{(\ell)}$ and $X_{\text{mlp}}^{(\ell)}$ into $\mathcal{X}_1$
    Collect o_proj input activations $X_o^{(\ell)}$
    $R_{2,\ell} \leftarrow \text{OPU}(R_{2,\ell}, X_o^{(\ell)})$
  end for
  $R_1 \leftarrow \text{OPU}(R_1, \mathcal{X}_1)$
end for
return $R_1, \{R_{2,\ell}\}_{\ell=1}^{L}$

3.3 Alternating orthogonal Procrustes updates

The prior objective 1 is non-smooth because the target z^i depends on the sign pattern of the rotated activation. Thus, we use an alternating procedure that updates the targets for a fixed rotation and then updates the rotation for the fixed targets. This yields a simple iterative algorithm in which each rotation update reduces to an orthogonal Procrustes problem.

Let $\tilde{x}^i = \frac{x^i}{\|x^i\|_2}$ denote the normalized activation vector. After fixing the targets, we can rewrite equation 1 by stacking normalized activations and targets into matrices $\tilde{X}, Z \in \mathbb{R}^{n\times d}$. The subproblem can then be written as

$$\min_{R\in O(d)} \left\| \tilde{X}R^\top - Z \right\|_F^2 \tag{3}$$

This problem is exactly the orthogonal Procrustes problem and has a known closed form solution [29]. Let

$$C = Z^\top \tilde{X}$$

with singular value decomposition

$$C = U\Sigma V^\top.$$

The optimal rotation is then given by

$$R \leftarrow UV^\top.$$

For convenience, we provide the derivation of this in Appendix C. We summarize this entire procedures with the operation

$$R = \text{OPU}(R, X),$$

where OPU stands for orthogonal Procrustes update.

A useful feature of this formulation is that it avoids gradient-based optimization over the orthogonal group. Instead of parameterizing R through a differentiable surrogate and performing gradient-based optimization, each update is solved exactly for the current targets.

Table 1: Activation quantization results of rotation-based methods on Llama-2 and Llama-3 models. We report average perplexity across WikiText-2, PTB, and C4, and average zero-shot accuracy across nine common sense reasoning tasks.

Bits (W-A-KV)	Method	Llama-2 7B		Llama-2 13B		Llama-3.2 3B		Llama-3 8B		Llama-3 70B	
		PPL↓	0-shot[9] ↑	PPL↓	0-shot[9] ↑	PPL↓	0-shot[9] ↑	PPL↓	0-shot[9] ↑	PPL↓	0-shot[9] ↑
16-16-16	FloatingPoint	16.88	61.16	20.85	64.28	10.9	61.43	8.92	66.04	6.19	72.70
4-4-16	QuaRot	20.34	57.89	24.82	62.36	13.75	55.64	11.06	61.12	13.26	57.53
	DFRot	**18.66**	57.93	23.50	**62.74**	13.76	55.66	10.96	61.52	11.70	60.72
	SpinQuant	19.97	58.16	23.75	62.50	13.65	56.20	11.13	61.80	9.31	65.43
	DartQuant	18.88	58.25	24.34	61.84	13.37	56.75	10.79	61.96	8.31	68.16
	Ours f.p. calib.	19.28	58.13	**22.78**	62.13	13.29	56.13	10.70	61.80	**8.25**	**68.17**
	Ours quant. calib.	18.77	**58.28**	23.24	62.43	**13.21**	**56.79**	**10.62**	**62.51**	8.29	67.63
4-4-4	QuaRot	21.08	57.45	25.17	62.16	14.23	55.75	11.31	60.82	13.53	57.59
	DFRot	19.18	57.78	23.57	**62.26**	14.24	54.74	11.19	60.89	12.02	59.59
	SpinQuant	20.69	57.62	24.05	61.62	13.98	55.43	11.28	60.95	9.06	65.52
	DartQuant	19.57	**58.02**	24.57	61.87	13.86	**55.79**	11.00	61.20	8.46	67.58
	Ours f.p. calib.	19.15	57.99	**23.02**	61.89	13.74	54.87	10.88	61.40	**8.40**	67.79
	Ours quant. calib.	**18.64**	57.89	23.12	62.24	**13.68**	55.77	**10.81**	**61.67**	8.49	**68.13**

3.4 Online mini-batch calibration

We combine our Procrustes formulation with a simple online calibration algorithm. Instead of solving for the rotation using all activations from the full calibration set at once, we process a small number of full sequences at a time. Given the current rotations $R_1, \{R_{2,\ell}\}_{\ell=1}^{L}$, we run the model on a mini-batch of calibration sequences, extract activation vectors at the calibration sites of interest, normalize each activation vector, compute the targets induced by the current rotation, accumulate the corresponding Procrustes statistics, and then immediately update the rotation.

An important consequence of the online formulation is that it enables quantization-aware rotation optimization. Because quantizing activations at one layer alters the hidden states passed to subsequent layers, the activation distribution seen by later layers during inference differs from the full-precision distribution. The online procedure allows us to account for this shift by optimizing each rotation using the quantized activations produced by earlier layers. Accordingly, during calibration we quantize the input to each linear layer after applying the current rotation so that optimization is matched to inference-time conditions.

Algorithm 1 summarizes the full online calibration procedure. For notational simplicity, the algorithm writes $R_{2,\ell}$ as a single matrix. In implementation, this matrix is constrained to be block diagonal across attention heads, as described in Appendix A. Thus, we solve a Procrustes problem per head. In addition, we do not need to store intermediate activations $X_{\text{attn}}^{(\ell)}, X_{\text{mlp}}^{(\ell)}$ across all transformer blocks as the cross covariance $Z^\top \tilde{X}$ can be computed per layer and accumulated. This sequence-wise mini-batch scheme allows us to calibrate rotations efficiently without materializing all token activations from the entire calibration set, while still retaining the benefits of closed-form Procrustes updates.

4 Experiments

We evaluate our method on the Llama-2 [1] and Llama-3 [2] model families, with model sizes ranging from 3B to 70B parameters. We compare against recent rotation-based methods for activation quantization, including QuaRot [20], SpinQuant [21], DartQuant [22], and DFRot [23]. We sample 128 calibration sequences of length 2,048 from WikiText-2 [30] and use this both for rotation calibration and weight quantization with GPTQ [13]. Activations are quantized using asymmetric per-token quantization, and, when applicable, the KV cache is quantized using asymmetric quantization with group size 128. Additional implementation details are provided in Appendix D.

We report perplexity (PPL) on WikiText-2 [30], Penn Treebank (PTB) [31], and C4 [32]. We also evaluate common-sense reasoning performance on nine downstream benchmarks: WinoGrande [33], SocialIQA [34], LAMBADA [35], MMLU [36], ARC-Easy, ARC-Challenge [37], HellaSwag [38], OpenBookQA [39], and PIQA [40].

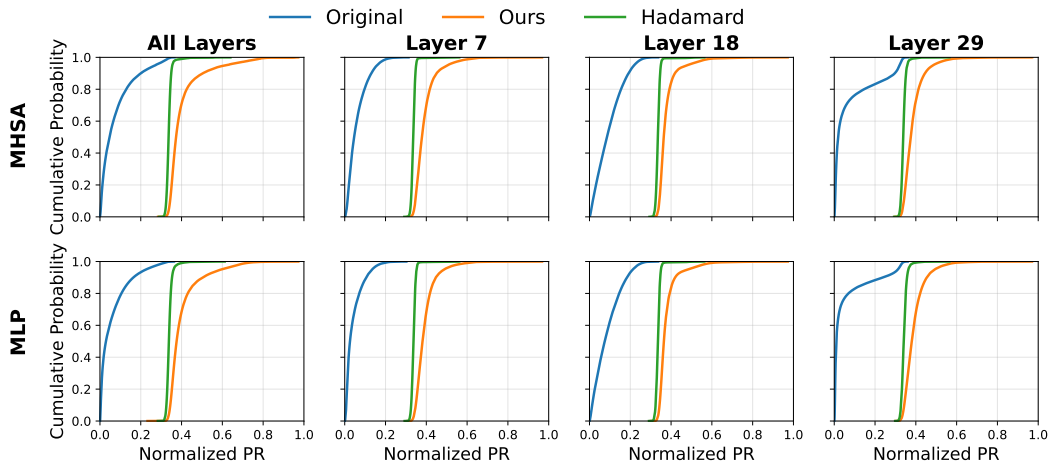

Figure 2: Empirical CDF of normalized participation ratio (PR) of activations from Llama-2 7B with different rotations.

4.1 Main results

We evaluate two quantization settings: 4-4-16, where weights and activations are quantized to 4 bits while the KV cache remains at 16 bits, and 4-4-4, where the KV cache is also quantized to 4 bits and present results in Table 1.

Across both 4-4-16 and 4-4-4 settings, our method achieves strong perplexity relative to prior rotation-based methods. The quantization-aware variant obtains the best average PPL for most Llama-3.2 3B and Llama-3 8B settings and remains competitive on Llama-2 7B/13B and Llama-3 70B. These gains are most consistent when calibration is performed with activation quantization enabled, supporting the value of matching calibration to inference-time activation distributions.

On nine zero-shot reasoning benchmarks, our method remains competitive with the strongest baselines and often improves the average score. Thus, the PPL improvements do not come from uniformly degraded downstream behavior. Importantly, these results are achieved without the end-to-end rotation training used by SpinQuant or the stored activation corpus required by offline activation-calibration methods.

4.2 Participation ratio analysis

The theory above suggests that effective rotations should move normalized activations toward the equal-magnitude regime that minimizes $\|R\tilde{x}\|_\infty$ on the unit sphere. To measure this behavior empirically, we use the normalized participation ratio [41],

$$\frac{\mathrm{PR}(y)}{d} = \frac{1}{d}\frac{\|y\|_2^4}{\|y\|_4^4} \in [1/d, 1].$$

PR can be interpreted as the effective fraction of coordinates carrying activation energy: it equals $1/d$ for a one-sparse vector and 1 when all coordinates have equal magnitude. Thus, larger normalized PR indicates that activation energy is more evenly distributed across dimensions, the same geometry targeted by our objective.

To evaluate this behavior empirically, we compare the normalized participation ratio, PR/d, of activations from the calibration set under three settings: no rotation, Hadamard rotations, and our optimized rotations and present the results in Figure 2.

Without rotation, the normalized PR is concentrated near zero, indicating that activation energy is highly concentrated in a small number of coordinates. Hadamard rotations substantially increase PR, with most activations attaining normalized PR values near 0.3, suggesting that random mixing already spreads activation energy across a larger fraction of the feature dimensions. However, our optimized rotations consistently shift the distribution further to the right, yielding higher PR values

Table 2: Comparison between online calibration and an offline stored-activation variant on Llama-3 8B.

Bits (W/A/KV)	Method	WikiText2↓	PTB↓	C4↓	Avg↓
4-4-16	Offline	7.32	12.71	11.99	10.67
	Online (f.p. calib.)	7.35	12.81	11.93	10.70
	Online (quant. calib.)	**7.31**	**12.65**	**11.89**	**10.62**
4-4-4	Offline	7.45	12.95	12.23	10.88
	Online (f.p. calib.)	7.47	13.03	12.15	10.88
	Online (quant. calib.)	**7.44**	**12.89**	**12.11**	**10.81**

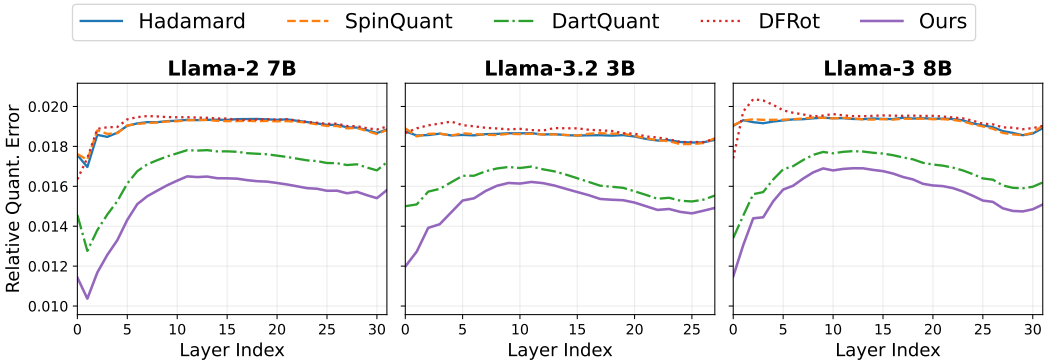

Figure 3: Layerwise activation quantization error for different rotation methods on Llama-2 7B, Llama-3.2 3B, and Llama-3 8B. We report the relative squared error between full-precision activations and their quantized reconstructions after applying each rotation.

across both MHSA and MLP activations. Moreover, our method shifts a noticeable fraction of activations into the higher-PR regime, with some activations reaching normalized PR values above 0.6. Since normalized PR measures the effective fraction of dimensions over which activation energy is distributed, this indicates that our rotations substantially reduce feature concentration relative to the original activations and the Hadamard baseline.

4.3 Quantization error analysis

To better understand the effect of the learned rotations, we directly measure activation quantization error across layers. For each model, we compare the relative squared quantization error

$$\frac{\|XR^\top - Q(XR^\top)\|_F^2}{\|X\|_F^2}$$

across rotations obtained by different methods. This analysis isolates the effect of the rotations on the activation distributions themselves, independent of downstream perplexity or task-level evaluation.

Figure 3 shows these results for Llama-2 7B, Llama-3.2 3B, and Llama-3 8B. Across all three models, our method consistently yields the lowest relative error across all layers. Compared with fixed Hadamard rotations and learned rotations from prior work, our method produces activations that are substantially easier to quantize, supporting the motivation of aligning normalized activations with hypercube corners to more evenly distribute activation magnitude across feature dimensions.

We also observe that DartQuant reduces quantization error relative to Hadamard, SpinQuant, and DFRot, but remains consistently above our method. This suggests that directly calibrating rotations toward quantization-friendly activation geometry provides a stronger reduction in quantization error than prior rotation objectives. These results help explain the perplexity improvements observed in Table 1: by reducing the per-layer distortion introduced by activation quantization, our rotations preserve the full-precision computation more effectively under low-bit inference.

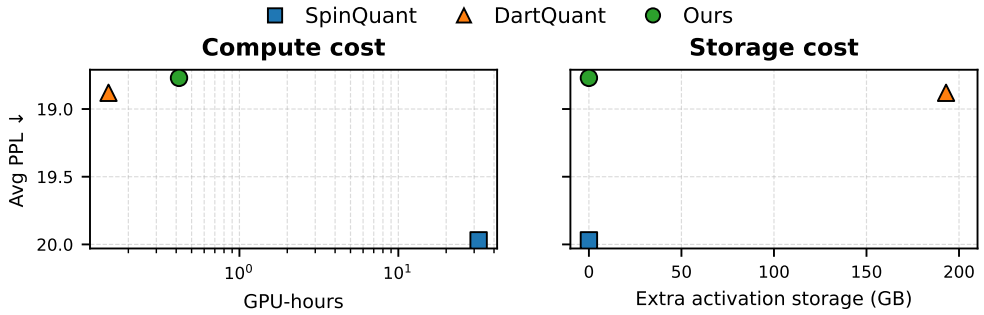

Figure 4: Calibration cost versus performance on Llama-2 7B under 4-4-16 quantization. Left: average PPL versus GPU-hours. Right: average PPL versus extra activation storage. Y-axis is inverted so upper-left is best.

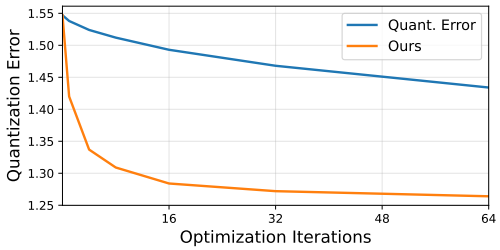

Figure 5: Comparison of quantization error per optimization step for different objectives.

Table 3: Average perplexity and common sense reasoning results for Llama-2 7B using different optimization objectives. Results shown represent performance with weights and activations quantized to 4 bits and KV cache left in 16 bits.

Objective	Avg. PPL ↓	Avg. Acc ↑
Quant. Error	19.39	57.88
Ours	18.77	58.28

4.4 Effect of online calibration

A key component of our method is its online optimization. Rather than first storing a large corpus of intermediate activations and then optimizing rotations from this fixed dataset, our method updates the rotations directly as calibration samples are processed. This avoids the storage overhead of offline activation collection and also enables calibration under the same quantized activation distributions encountered at inference time.

To isolate the effect of this design choice, we compare our online procedure with an offline variant that stores all calibration activations to disk and then samples them i.i.d. during optimization. To ensure a fair comparison, we fix the number of activations used per Procrustes update and the number of activation steps. We evaluate both variants on Llama-3 8B and present results in Table 2. The offline variant performs comparably to our online method when calibration is performed in full precision, indicating that the online updates closely approximate the behavior of optimization over stored i.i.d. activations. This suggests that our online formulation preserves the benefits of activation-based rotation optimization while avoiding the need to materialize the activation corpus. Lastly, we observe that enabling quantization during calibration improves performance supporting the motivation for quantization-aware calibration.

4.5 Calibration efficiency.

Figure 4 compares the calibration cost–performance tradeoff for Llama-2 7B under 4-4-16 quantization. SpinQuant learns rotations through gradient-based optimization and, in our setup, requires 8 hours on 4 RTX A6000 GPUs (32 GPU-hours). DartQuant reduces calibration time to 9 minutes on a single RTX A6000, but does so by storing approximately 193GB of intermediate activations. Our method instead performs closed-form online calibration, requiring 25 minutes on a single RTX A6000 (0.42 GPU-hours) and no offline activation storage. Thus, our method occupies a favorable point in the tradeoff: it avoids SpinQuant's high compute cost and DartQuant's large storage overhead, while achieving the best average perplexity among these methods.

4.6 Corner alignment optimization vs quantization error optimization

Our proposed method consists of two main components: a corner alignment objective that can be interpreted as an orthogonal Procrustes problem and an online optimization framework. While minimizing the corner alignment objective reduces the overall quantization error, the objective can be formulated explicitly using quantization error while still being solved via iterative orthogonal Procrustes updates. This can be seen by letting $\tilde{X}_Q = Q(XR^\top)$, where $Q(\cdot)$ represents the quantization function, and forming the below optimization

$$\min_{R \in O(d)} \left\| XR^\top - \tilde{X}_Q \right\|_F^2 .$$

This problem has the same form as our corner alignment objective and can thus be solved similarly. Because of this, it is necessary to explore whether the effectiveness of the proposed method is a result of the proposed corner alignment objective or if it is primarily due to the online framework combined with closed-form Procrustes updates. We investigate this by replacing the corner objective inside of our online optimization framework with the objective defined above. We report average relative quantization error per optimization step in Figure 5 and perplexity and average common sense reasoning performance for Llama-2 7B in Table 3. As can be seen, optimizing the corner objective reduces quantization error much more quickly. This reduction in quantization error directly translates to improved perplexity and common sense reasoning performance, suggesting that the performance of our method is due to both the corner alignment objective and the online optimization framework.

5 Conclusion and limitations

We introduced an efficient rotation-based calibration method for low-bit LLM activation quantization. By aligning normalized activations with the vertices of an inscribed hypercube, our method encourages activation energy to be distributed more evenly across feature dimensions, reducing the effect of outlier coordinates that dominate asymmetric quantization error. Our work avoids specialized optimizers for gradient-based optimization on the orthogonal group while also avoiding the computational and storage costs of prior works. Experiments across multiple LLM families and benchmarks show that this simple calibration procedure provides competitive quantized performance while substantially reducing calibration cost and storage overhead. These results suggest that closed-form, online rotation calibration is a practical direction for scalable post-training quantization of large language models. By reducing costs associated with rotation calibration, we improve the accessibility and ease of use of LLMs. However, if these models are misused, this could have negative societal impact.

Limitations Our method adds a calibration step and requires optimization over a small set of calibration data, so performance may depend on the calibration data distribution. In addition, our experiments focus primarily on perplexity and common downstream benchmarks; broader evaluation of generation quality and latency remains important future work. We additionally fix the weight quantization method as GPTQ in all experiments. Follow-up experiments are necessary to evaluate whether performance gains hold when combined with various weight quantization approaches. Finally, our experiments focus on 4-bit weight/activation quantization due to the availability of efficient 4-bit kernels. Future work should investigate whether these performance improvements persist under more severe weight and activation quantization.

Acknowledgements

This work was supported in part by the National Center on Generative AI for Uplifting STEM+C Education (GENIUS Center) and by the United States Air Force under Contract No. FA8750-23-C-0518. The authors also gratefully acknowledge the computing resources provided by the NVIDIA Academic Grant Program.

References

[1] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

[2] Aaron Grattafiori, Abhimanyu Dubey, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[4] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

[5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sashank Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24(1), January 2023. ISSN 1532-4435.

[6] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

[7] Elias Frantar and Dan Alistarh. Sparsegpt: massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[8] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PxoFut3dWW.

[9] Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-pruner: On the structural pruning of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=J8Ajf9WfXP.

[10] Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LNYIUouhdt.

[11] Zhihang Yuan, Yuzhang Shang, Yue Song, Dawei Yang, Qiang Wu, Yan Yan, and Guangyu Sun. ASVD: Activation-aware singular value decomposition for compressing large language models, 2025. URL https://openreview.net/forum?id=HyPofygOCT.

[12] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

[13] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tcbBPnfwxS.

[14] 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 llm compression and acceleration. In *MLSys*, 2024.

[15] 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 Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 38087–38099. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

[16] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[17] Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: efficient and affordable post-training quantization for large-scale transformers. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[18] Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8Wuvhh0LYW.

[19] Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. Outlier suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1648–1665, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.102. URL https://aclanthology.org/2023.emnlp-main.102/.

[20] Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=dfqsW38v1X.

[21] Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: LLM quantization with learned rotations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ogO6DGE6FZ.

[22] Yuantian Shao, Yuanteng Chen, Peisong Wang, Jianlin Yu, Jing Lin, Yiwu Yao, Zhihui Wei, and Jian Cheng. Dartquant: Efficient rotational distribution calibration for LLM quantization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=LfcfwlLCHM.

[23] Jingyang Xiang and Sai Qian Zhang. DFRot: Achieving outlier-free and massive activation-free for rotated LLMs with refined rotation. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=WzGypILLDb.

[24] Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. Squeezellm: dense-and-sparse quantization. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[25] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-bit quantization of large language models with guarantees. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xrk9g5vcXR.

[26] Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP$\#$: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=9BrydUVcoe.

[27] Albert Tseng, Qingyao Sun, David Hou, and Christopher De Sa. QTIP: Quantization with trellises and incoherence processing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=7sdkLVuYCU.

[28] Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. Duquant: distributing outliers via dual transformation makes stronger quantized llms. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

[29] Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1–10, 1966. URL https://api.semanticscholar.org/CorpusID:121676935.

[30] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

[31] Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The penn treebank: annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, page 114–119, USA, 1994. Association for Computational Linguistics. ISBN 1558603573. doi: 10.3115/1075812.1075835. URL https://doi.org/10.3115/1075812.1075835.

[32] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435.

[33] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, August 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

[34] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454/.

[35] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. openai blog (2019). *URL: https://d4mucfpksywv. cloudfront. net/better-language-models/language-models. pdf*, 2024.

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

[37] Michael Boratko, Harshit Padigela, Divyendra Mikkilineni, Pritish Yuvraj, Rajarshi Das, Andrew McCallum, Maria Chang, Achille Fokoue-Nkoutche, Pavan Kapanipathi, Nicholas Mattei, Ryan Musa, Kartik Talamadupula, and Michael Witbrock. A systematic classification of knowledge, reasoning, and context within the ARC dataset. In Eunsol Choi, Minjoon Seo, Danqi Chen, Robin Jia, and Jonathan Berant, editors, *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 60–70, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-2607. URL https://aclanthology.org/W18-2607/.

[38] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

[39] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

[40] Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439, Apr. 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

[41] R. J. Bell and P. Dean. Atomic vibrations in vitreous silica. *Discuss. Faraday Soc.*, 50:55–61, 1970. doi: 10.1039/DF9705000055. URL http://dx.doi.org/10.1039/DF9705000055.

[42] Biao Zhang and Rico Sennrich. *Root mean square layer normalization*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[43] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C), February 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

[44] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[45] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile

Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

[46] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Appendix

A Rotation invariance in LLMs

Let $\hat{X} = XR$, where $X \in \mathbb{R}^{T\times d}$ denotes the activations of T tokens and $R \in \mathbb{R}^{d\times d}$ is an orthogonal matrix. If $\hat{X}$ is passed to a linear layer with weight matrix $W \in \mathbb{R}^{d_o\times d}$, then the original computation can be preserved by absorbing the inverse rotation into the weight matrix:

$$XW^\top = \hat{X}(R^\top W^\top).$$

Thus, rotating the activations is equivalent to replacing W by a rotated weight matrix, so no additional floating-point operations are required at inference for these mergeable rotations.

In pre-norm transformers, the residual stream is typically normalized before entering the self-attention and MLP blocks. For RMSNorm [42], this does not break rotational invariance, since the normalization depends only on the vector norm. In practice, the learned RMSNorm scale is first fused into the adjacent linear weights, after which the rotation can be absorbed exactly into the surrounding weight matrices. This is the same computational-invariance mechanism used by prior rotation-based quantization methods.

Following prior work [20, 21, 22, 23], we optimize a single rotation shared across all layers denoted as R_1 as well as a layer specific rotation denoted as $R_{2,\ell}$. The rotation R_1 acts on the residual stream and is shared across all transformer blocks. It is applied to the inputs of the self-attention and MLP sublayers, and its inverse is absorbed into the corresponding input projections. Because the residual stream is propagated through skip connections, the same R_1 must be used throughout the network; correspondingly, the output projections that write back to the residual stream absorb the matching inverse transform. In our notation, this means that R_1 is merged into the weights associated with the attention and MLP inputs, while the inverse transform is absorbed into the output projections W_O and W_{down}.

We additionally use a layer-specific rotation $R_{2,\ell}$ inside each attention block. This rotation is applied to the activations input to the output projection W_O and can therefore be paired with a corresponding transform on the value path. Concretely, $R_{2,\ell}$ is absorbed into the value projection W_V, while its inverse is absorbed into W_O, yielding an equivalent full-precision computation but a more quantization-friendly activation distribution at the input to the output projection. As these rotations do not need to be shared across layers, each $R_{2,\ell}$ rotation is learned separately for each transformer block. Lastly, due to the self-attention operation being performed across multiple heads, $R_{2,\ell}$ is constrained to be block diagonal with block size $d_h \times d_h$.

The rotations above are the only ones we explicitly optimize and merge into the model weights. In addition, following QuaRot[20] and SpinQuant[21], we optionally use online Hadamard transforms to further reduce outliers in locations where mergeable rotations are not sufficient or cannot be absorbed as conveniently. We denote these fixed transforms by R_3 and R_4. The transform R_3 is applied after the rotary positional embeddings (RoPE)[43] within self-attention to improve KV-cache quantization, while R_4 is applied before the down projection to improve low-bit activation quantization inside the MLP. Because Hadamard transforms admit efficient fast implementations, they can be applied online during inference with modest overhead.

B From quantization error to the corner objective derivation

We now provide a theoretical justification for aligning activations with the corners of the inscribed hypercube. The argument proceeds in three steps: (i) the mean-squared error of both uniform symmetric and asymmetric (zeropoint) quantizers is controlled by the ratio $\|x\|_\infty^2/\|x\|_2^2$; (ii) on the unit sphere, this ratio attains its minimum exactly at the hypercube corners; (iii) the corner-distance objective in equation 1 is equivalent to a smooth ℓ_1-maximization that serves as a tight, differentiable proxy for minimizing $\|x\|_\infty$.

Quantization error of a per-tensor symmetric uniform quantizer. Consider a symmetric b-bit uniform quantizer applied to an activation vector $x \in \mathbb{R}^d$ with per-tensor step size determined by the

absolute maximum coordinate,

$$Q_{\mathrm{sym}}(x) \;=\; \Delta_{\mathrm{sym}}(x)\,\mathrm{round}\!\left(\frac{x}{\Delta_{\mathrm{sym}}(x)}\right), \qquad \Delta_{\mathrm{sym}}(x) \;=\; \frac{\|x\|_\infty}{2^{b-1}-1}.$$

Each coordinate incurs a rounding error bounded in magnitude by $\Delta_{\mathrm{sym}}(x)/2$, so the total mean-squared quantization error satisfies

$$\|Q_{\mathrm{sym}}(x) - x\|_2^2 \;\le\; \frac{d\,\Delta_{\mathrm{sym}}(x)^2}{4} \;=\; \frac{d}{4(2^{b-1}-1)^2}\,\|x\|_\infty^2. \tag{4}$$

The signal-to-quantization-noise ratio (SQNR) is therefore lower bounded by

$$\mathrm{SQNR}(x) \;=\; \frac{\|x\|_2^2}{\|Q_{\mathrm{sym}}(x) - x\|_2^2} \;\ge\; \frac{4(2^{b-1}-1)^2}{d}\cdot\frac{1}{\rho(x)}, \qquad \rho(x) \;:=\; \frac{\|x\|_\infty^2}{\|x\|_2^2}. \tag{5}$$

The dynamic-range ratio $\rho(x) \in [1/d, 1]$ is the only data-dependent quantity in the bound: minimizing ρ over the unit sphere directly maximizes the worst-case SQNR.

Quantization error of a per-tensor asymmetric (zeropoint) quantizer. A zeropoint quantizer [44] maps the affine range $[\min_j x_j,\, \max_j x_j]$ onto the full unsigned integer grid $\{0, \ldots, 2^b - 1\}$ via a scale and a shift,

$$Q_{\mathrm{zp}}(x) \;=\; \Delta_{\mathrm{zp}}(x)\left(\mathrm{round}\!\left(\tfrac{x}{\Delta_{\mathrm{zp}}(x)} + \zeta(x)\right) - \zeta(x)\right), \quad \Delta_{\mathrm{zp}}(x) \;=\; \frac{\max_j x_j - \min_j x_j}{2^b - 1},$$

where $\zeta(x) \in \mathbb{Z}$ is the integer zeropoint that aligns $\min_j x_j$ with the lower end of the grid. Each coordinate again incurs a rounding error bounded by $\Delta_{\mathrm{zp}}(x)/2$, yielding

$$\|Q_{\mathrm{zp}}(x) - x\|_2^2 \;\le\; \frac{d\,\Delta_{\mathrm{zp}}(x)^2}{4} \;=\; \frac{d}{4(2^b-1)^2}\left(\max_j x_j - \min_j x_j\right)^2. \tag{6}$$

The data-dependent quantity is now the signed range $\max_j x_j - \min_j x_j$ rather than $\|x\|_\infty$. These two are tied by the elementary inequality

$$\max_j x_j - \min_j x_j \;\le\; 2\|x\|_\infty, \tag{7}$$

with equality whenever x has at least one coordinate of each sign achieving the maximum absolute value—in particular at every sign-mixed vertex of the inscribed hypercube. Combining (6) and (7) gives

$$\|Q_{\mathrm{zp}}(x) - x\|_2^2 \;\le\; \frac{d}{(2^b-1)^2}\,\|x\|_\infty^2, \tag{8}$$

matching the symmetric bound (4).

Tightness on the unit sphere. A keen reader may worry that (7) can be loose: the bound upper-bounds the signed range by $2\|x\|_\infty$, but the two quantities need not share the same minimizer on the unit sphere. We address this concern by quantifying both sides directly. For $x \in \mathbb{S}^{d-1}$,

$$0 \;\le\; \max_j x_j - \min_j x_j \;\le\; \sqrt{2}, \qquad \frac{1}{\sqrt{d}} \;\le\; \|x\|_\infty \;\le\; 1.$$

The signed range attains its minimum of 0 at the all-same-sign directions $\pm\mathbf{1}/\sqrt{d}$, while $\|x\|_\infty$ attains its minimum $1/\sqrt{d}$ at every vertex of the inscribed hypercube. The two surrogates therefore disagree on the global optimum: minimizing $\|x\|_\infty$ drives x to an arbitrary hypercube corner, whereas minimizing the signed range would prefer the two specific same-sign corners.

The disagreement, however, vanishes with dimension. At any sign-mixed hypercube corner—the generic fixed point of the corner objective—the signed range equals $2/\sqrt{d}$, while the unreachable global minimum is 0. The absolute gap is $O(1/\sqrt{d})$, and the achieved value lies within a fraction $\sqrt{2/d}$ of the worst-case range $\sqrt{2}$. For transformer-scale hidden dimensions this gap is negligible: at $d = 4096$, sign-mixed corners achieve a range of $2/\sqrt{4096} \approx 0.031$, recovering more than 97% of the available range reduction on the unit sphere. The factor-of-two slack in (7) is loose at axis poles—which the optimization moves away from—and asymptotically tight at the corners it converges to. Minimizing $\|x\|_\infty$ therefore controls the MSE bound for both quantizers simultaneously, with a zeropoint-side suboptimality that decays as $1/\sqrt{d}$ and is quantitatively immaterial in the regime of interest. The remainder of this section may be read with $\|x\|_\infty$ as the single quantity to minimize, with the understanding that the analysis applies verbatim to symmetric quantization and, up to vanishing dimensional corrections, to asymmetric quantization as well.

Corners minimize the dynamic-range ratio. An orthogonal rotation $R \in O(d)$ preserves ℓ_2 norms, so $\|R\tilde{x}\|_2 = 1$ for any normalized activation $\tilde{x}$. Rotation therefore operates entirely through the numerator of ρ:

$$\rho(R\tilde{x}) \;=\; \|R\tilde{x}\|_\infty^2.$$

The following standard fact identifies the minimizers.

Proposition 1 *For any $y \in \mathbb{R}^d$ with $\|y\|_2 = 1$, $\|y\|_\infty^2 \geq 1/d$, with equality if and only if y is a vertex of the hypercube inscribed in the unit sphere, i.e. $y = \frac{1}{\sqrt{d}}(\pm 1, \ldots, \pm 1)$.*

Combined with (5), this shows that hypercube corners are the directions on the sphere at which the quantization error bound attains its minimum value

$$\|Q_{\mathrm{sym}}(y) - y\|_2^2 \leq \frac{1}{4(2^{b-1}-1)^2}.$$

For comparison, axis poles $\pm e_j$ achieve the worst case $\rho = 1$, in which case the bound reduces to

$$\|Q_{\mathrm{sym}}(y) - y\|_2^2 \leq \frac{d}{4(2^{b-1}-1)^2}.$$

which is d larger than the corner regime.

Dualization: an ℓ_1 surrogate for ℓ_∞. Although Proposition 1 identifies the ideal target, $\|y\|_\infty$ is non-smooth and its subgradient concentrates on a single coordinate, making direct minimization over $O(d)$ ill-conditioned. We instead work with the dual norm via its variational representation. Recall that ℓ_1 and ℓ_∞ are Hölder conjugates, with

$$\|y\|_1 \;=\; \max_{s \in \{-1,+1\}^d} \langle s,\, y \rangle \;=\; \langle \mathrm{sign}(y),\, y \rangle. \tag{9}$$

On the unit sphere, Cauchy–Schwarz gives

$$\|y\|_1 \;\leq\; \sqrt{d}\,\|y\|_2 \;=\; \sqrt{d}, \tag{10}$$

with equality if and only if $|y_1| = \cdots = |y_d| = 1/\sqrt{d}$, i.e., y is a hypercube corner. Thus, maximizing $\|R\tilde{x}\|_1$ has the same set of per-sample maximizers on the unit sphere as minimizing $\|R\tilde{x}\|_\infty$, while providing dense subgradient information across coordinates. Substituting (9) introduces auxiliary sign variables $s^i \in \{-1,+1\}^d$ and yields the equivalent problem

$$\max_{R \in O(d)} \sum_i \max_{s^i \in \{-1,+1\}^d} \langle s^i,\, R\tilde{x}^i \rangle. \tag{11}$$

Fixing the sign variables makes the objective linear in R, and the resulting maximization over $O(d)$ reduces to an orthogonal Procrustes problem. This alternating maximization—updating $s^i \leftarrow \mathrm{sign}(R\tilde{x}^i)$ followed by a closed-form Procrustes update for R—is exactly the procedure used in Section 3.3.

Equivalence to the corner-distance objective. The corner-distance objective in Equation 1 is the rescaled form of the dualized problem (11) in disguise: identifying $z^i = s^i/\sqrt{d}$ recovers the targets defined coordinatewise as $z_j^i = \frac{1}{\sqrt{d}}\,\mathrm{sign}((R\tilde{x}^i)_j)$. To see this explicitly, expand the squared distance:

$$\begin{aligned}
\|R\tilde{x}^i - z^i\|_2^2 &= \|R\tilde{x}^i\|_2^2 + \|z^i\|_2^2 - 2\,\langle R\tilde{x}^i,\, z^i \rangle \\
&= 2 \;-\; \frac{2}{\sqrt{d}} \sum_{j=1}^{d} (R\tilde{x}^i)_j\,\mathrm{sign}((R\tilde{x}^i)_j) \\
&= 2 \;-\; \frac{2}{\sqrt{d}}\,\|R\tilde{x}^i\|_1,
\end{aligned}$$

where the last equality uses the variational identity (9) evaluated at the optimal sign vector $s^i = \mathrm{sign}(R\tilde{x}^i)$. Summing over the calibration set yields

$$\sum_i \|R\tilde{x}^i - z^i\|_2^2 \;=\; 2N \;-\; \frac{2}{\sqrt{d}} \sum_i \|R\tilde{x}^i\|_1, \tag{12}$$

so that minimizing the corner distance is exactly equivalent to maximizing the average ℓ_1 norm of the rotated, normalized activations—the outer problem of (11) after the inner sign maximization has been solved in closed form. By the Cauchy–Schwarz bound (10) and Proposition 1, this is a tight proxy for jointly minimizing $\|R\tilde{x}^i\|_\infty$ for each i, and hence for minimizing the upper bound (6) on quantization MSE in expectation across the calibration distribution.

C Procrustes update derivation

First, note that Equation 3 can be rewritten as

$$\min_{R \in O(d)} \left\| \tilde{X}R^\top - Z \right\|_F^2 = \min_{R \in O(d)} \|\tilde{X}\|_F^2 - 2\langle \tilde{X}R^\top, Z \rangle_F + \|Z\|_F^2, \tag{13}$$

where we have used the orthogonality of R to obtain $\|\tilde{X}R^\top\|_F^2 = \|\tilde{X}\|_F^2$. Since the first and last terms are independent of R, the optimization is equivalent to

$$\max_{R \in O(d)} \langle \tilde{X}R^\top, Z \rangle_F = \max_{R \in O(d)} \langle R, Z^\top \tilde{X} \rangle_F \tag{14}$$

$$= \max_{R \in O(d)} \mathrm{Tr}(R^\top Z^\top \tilde{X}). \tag{15}$$

Let $Z^\top \tilde{X} = U\Sigma V^\top$ denote its singular value decomposition. Then

$$\mathrm{Tr}(R^\top Z^\top \tilde{X}) = \mathrm{Tr}(R^\top U\Sigma V^\top) \tag{16}$$

$$= \mathrm{Tr}(V^\top R^\top U\Sigma) \tag{17}$$

$$\leq \mathrm{Tr}(\Sigma), \tag{18}$$

where the inequality follows from the orthogonality of $V^\top R^\top U$. The upper bound is attained when $V^\top R^\top U = I$ which occurs precisely when

$$R = UV^\top.$$

D Implementation details

Implementation details. We generate optimized rotations using the publicly available implementations of SpinQuant [21], DFRot [23], and DartQuant [22]. For Llama-3 70B, we use the publicly available rotations for SpinQuant which are trained on a larger corpus of calibration data. All methods are evaluated with a common evaluation pipeline adapted from the public DartQuant codebase. Unless otherwise stated, calibration is performed on the same 128 WikiText2 [30] samples for all methods to ensure a fair comparison. Since DFRot optimizes its rotation using a single calibration sample, we use the first sample from this shared calibration set.

For weight quantization, quantization scales are computed per output row. For activation and KV-cache quantization, we follow the DartQuant evaluation setting and use clipping ratios of 0.9 and 1.0, respectively. To evaluate QuaRot [20] within our rotation framework, we replace the learned rotations R_1 and $R_{2,\ell}$ with random Hadamard rotations, implemented using the Quip# Hadamard routines [26]. Our calibration procedure uses batch size 1 for all models, except for Llama-2 7B and Llama-3 8B, where we use batch size 2. During calibration, we initialize our rotations as hadamard matrices.

Experiments on Llama-2 7B, 13B, Llama-3.2 3B, and Llama-3 8B are all completed on servers with 4 RTX A6000 GPUs with calibration performed on a single GPU unless otherwise stated. For SpinQuant, we use all 4 GPUs during calibration. Llama-3 70B experiments are all completed using a single RTX Pro 6000 Blackwell GPU.

Table 4: Full perplexity results for Llama-2 7B

Bits (W/A/KV)	Method	WikiText2↓	PTB↓	C4↓	Avg↓
4-4-16	QuaRot	6.04	46.79	8.18	20.34
	DFRot	5.97	41.91	8.10	18.66
	SpinQuant	6.02	45.72	8.16	19.97
	DartQuant	5.96	42.62	8.07	18.88
	Ours f.p. calib.	5.94	43.86	8.04	19.28
	Ours quant. calib.	5.93	42.34	8.04	18.77
4-4-4	QuaRot	6.11	48.86	8.27	21.08
	DFRot	6.03	43.33	8.18	19.18
	SpinQuant	6.09	47.70	8.27	20.69
	DartQuant	6.02	44.52	8.17	19.57
	Ours f.p. calib.	6.00	43.33	8.13	19.15
	Ours quant. calib.	6.00	41.74	8.17	18.64

Table 5: Full common sense reasoning results for Llama-2 7B

Bits (W/A/KV)	Method	WG	SIQA	LAMB	MMLU	ARC-E	ARC-C	HS	OBQA	PIQA	Avg
4-4-16	QuaRot	67.09	43.55	72.04	34.6	70.58	42.66	73.31	39.6	77.58	57.89
	DFRot	67.17	44.11	71.96	34.75	69.81	42.41	73.0	40.8	77.37	57.93
	SpinQuant	66.93	43.35	72.04	34.70	71.80	42.41	73.09	41.60	77.48	58.16
	DartQuant	65.11	44.06	70.62	37.07	72.60	44.11	73.32	40.20	77.20	58.25
	Ours f.p. calib.	66.22	44.52	71.76	33.21	71.09	43.0	73.68	41.8	77.86	58.13
	Ours quant. calib.	66.69	44.88	72.68	33.56	71.04	42.41	73.41	42.40	77.48	58.28
4-4-4	QuaRot	66.61	43.86	70.77	34.44	69.49	41.81	72.86	40.2	77.04	57.45
	DFRot	66.61	44.58	71.45	34.55	69.40	42.06	72.73	42.2	76.33	57.78
	SpinQuant	65.59	44.17	71.1	35.07	70.54	40.53	72.79	40.8	77.97	57.62
	DartQuant	65.11	42.84	71.01	36.3	71.46	43.26	73.41	41.0	77.8	58.02
	Ours f.p. calib.	67.48	44.42	71.63	34.07	70.37	41.13	73.23	42.0	77.58	57.99
	Ours quant. calib.	67.88	44.11	70.79	34.33	70.33	42.75	72.81	40.4	77.64	57.89

E Full results

Below in Tables 4 to 13 we present the full results used to populate Table 1. We present perplexity results on WikiText2 [30], Penn TreeBank (PTB) [31], and C4 [32]. For common sense reasoning, we evaluate on WinoGrande (WG) [33], SocialIQA (SIQA) [34], LAMBADA (LAMB) [35], MMLU [36], ARC-Easy (ARC-E), ARC-Challenge (ARC-C) [37], HellaSwag (HS) [38], OpenBookQA (OBQA) [39], and PIQA [40].

Table 6: Full perplexity results for Llama-2 13B

Bits (W/A/KV)	Method	WikiText2↓	PTB↓	C4↓	Avg↓
4-4-16	QuaRot	5.29	61.78	7.38	24.82
	DFRot	5.25	57.93	7.31	23.50
	SpinQuant	5.32	58.58	7.36	23.75
	DartQuant	5.29	60.35	7.37	24.34
	Ours f.p. calib.	5.25	55.78	7.31	22.78
	Ours quant. calib.	5.25	57.13	7.33	23.24
4-4-4	QuaRot	5.33	62.76	7.43	25.17
	DFRot	5.29	58.06	7.36	23.57
	SpinQuant	5.37	59.34	7.44	24.05
	DartQuant	5.34	60.95	7.43	24.57
	Ours f.p. calib.	5.29	56.41	7.37	23.02
	Ours quant. calib.	5.30	57.67	7.40	23.12

Table 7: Full common sense reasoning results for Llama-2 13B

Bits (W/A/KV)	Method	WG	SIQA	LAMB	MMLU	ARC-E	ARC-C	HS	OBQA	PIQA	Avg
4-4-16	QuaRot	68.51	45.7	75.78	47.08	74.92	48.12	76.79	44.6	79.76	62.36
	DFRot	71.82	46.52	75.59	46.3	75.97	48.21	76.93	44.6	78.73	62.74
	SpinQuant	71.27	46.01	74.85	47.71	75.21	47.61	76.26	44.2	79.38	62.5
	DartQuant	71.51	43.71	75.41	44.19	74.07	47.7	77.19	43.8	78.94	61.84
	Ours f.p. calib.	69.77	44.98	74.54	48.1	74.75	47.95	75.9	44.0	79.16	62.13
	Ours quant. calib.	70.8	46.21	75.16	47.17	75.21	48.55	77.45	42.4	78.89	62.43
4-4-4	QuaRot	69.14	45.55	75.33	46.08	75.04	48.81	76.78	43.4	79.33	62.16
	DFRot	69.93	45.8	75.02	45.86	75.88	48.46	77.26	43.2	78.94	62.26
	SpinQuant	69.38	45.04	74.44	46.22	74.66	46.59	76.52	43.4	78.29	61.62
	DartQuant	69.93	44.47	74.85	47.03	74.28	45.82	76.75	44.8	78.89	61.87
	Ours f.p. calib.	71.27	44.27	74.73	47.33	74.37	46.93	75.64	43.4	79.05	61.89
	Ours quant. calib.	70.72	46.01	74.91	46.72	75.17	47.1	77.18	43.8	78.51	62.24

Table 8: Full perplexity results for Llama-3.2 3B

Bits (W/A/KV)	Method	WikiText2↓	PTB↓	C4↓	Avg↓
4-4-16	QuaRot	9.62	16.65	14.99	13.75
	DFRot	9.65	16.63	15.01	13.76
	SpinQuant	9.51	16.58	14.86	13.65
	DartQuant	9.35	16.16	14.59	13.37
	Ours f.p. calib.	9.25	16.19	14.43	13.29
	Ours quant. calib.	9.19	16.10	14.33	13.21
4-4-4	QuaRot	9.91	17.28	15.50	14.23
	DFRot	9.94	17.30	15.48	14.24
	SpinQuant	9.73	16.98	15.24	13.98
	DartQuant	9.63	16.91	15.05	13.86
	Ours f.p. calib.	9.50	16.83	14.90	13.74
	Ours quant. calib.	9.49	16.72	14.84	13.68

Table 9: Full common sense reasoning results for Llama-3.2 3B

Bits (W/A/KV)	Method	WG	SIQA	LAMB	MMLU	ARC-E	ARC-C	HS	OBQA	PIQA	Avg
4-4-16	QuaRot	62.12	43.86	62.7	46.2	64.6	38.91	68.93	39.0	74.48	55.64
	DFRot	62.51	44.73	61.13	45.7	63.89	41.04	68.42	40.0	73.5	55.66
	SpinQuant	63.93	44.27	63.3	46.54	66.41	40.36	68.48	38.0	74.54	56.20
	DartQuant	64.72	44.11	63.63	46.23	66.71	42.58	69.36	38.8	74.65	56.75
	Ours f.p. calib.	64.4	44.98	62.53	44.7	64.10	39.76	69.26	40.0	75.41	56.13
	Ours quant. calib.	65.19	44.37	64.41	46.01	66.50	40.78	68.75	70.6	74.54	56.79
4-4-4	QuaRot	64.09	45.04	61.5	44.06	64.77	40.19	68.0	40.0	74.10	55.75
	DFRot	63.3	44.32	59.27	44.19	62.79	38.4	68.17	37.60	74.59	54.74
	SpinQuant	63.14	43.45	62.10	44.75	64.44	40.36	68.16	37.40	75.03	55.43
	DartQuant	62.98	42.99	62.99	44.32	66.04	42.15	68.20	38.40	74.05	55.79
	Ours f.p. calib.	61.80	44.37	60.78	42.27	62.33	40.10	68.43	39.80	73.94	54.87
	Ours quant. calib.	64.25	44.78	62.31	44.75	63.09	39.85	68.72	39.60	74.59	55.77

Table 10: Full perplexity results for Llama-3 8B

Bits (W/A/KV)	Method	WikiText2↓	PTB↓	C4↓	Avg↓
4-4-16	QuaRot	7.62	13.21	12.34	11.06
	DFRot	7.58	12.91	12.38	10.96
	SpinQuant	7.64	13.39	12.37	11.13
	DartQuant	7.42	12.92	12.04	10.79
	Ours f.p. calib.	7.35	12.81	11.93	10.70
	Ours quant. calib.	7.31	12.65	11.89	10.62
4-4-4	QuaRot	7.78	13.53	12.62	11.31
	DFRot	7.74	13.22	12.61	11.19
	SpinQuant	7.75	13.52	12.58	11.28
	DartQuant	7.54	13.19	12.28	11.00
	Ours f.p. calib.	7.47	13.03	12.15	10.88
	Ours quant. calib.	7.44	12.89	12.11	10.81

Table 11: Full common sense reasoning results for Llama-3 8B

Bits (W/A/KV)	Method	WG	SIQA	LAMB	MMLU	ARC-E	ARC-C	HS	OBQA	PIQA	Avg
4-4-16	QuaRot	66.85	44.52	69.88	54.59	74.03	46.33	75.28	41.60	76.99	61.12
	DFRot	69.93	45.24	71.78	55.12	73.91	45.9	75.54	40.0	76.28	61.52
	SpinQuant	68.19	44.47	69.75	54.93	76.01	48.21	75.62	41.40	77.58	61.80
	DartQuant	68.75	45.04	71.2	54.8	73.36	47.78	75.26	43.60	77.86	61.96
	Ours f.p. calib.	70.56	45.04	70.17	55.53	72.98	47.10	75.89	42.40	76.61	61.80
	Ours quant. calib.	69.46	44.88	70.13	55.75	75.76	48.21	77.78	44.80	77.80	62.51
4-4-4	QuaRot	70.4	44.68	69.11	53.14	73.02	44.45	74.63	41.20	76.71	60.82
	DFRot	70.64	43.76	71.14	53.42	72.9	43.52	74.79	41.20	76.66	60.89
	SpinQuant	68.03	45.24	69.45	52.75	72.35	47.35	74.34	41.80	77.26	60.95
	DartQuant	68.03	45.39	70.58	53.35	73.53	46.76	74.89	41.0	77.31	61.20
	Ours f.p. calib.	70.96	44.32	69.75	54.17	72.81	46.50	75.16	41.60	77.31	61.40
	Ours quant. calib.	68.90	45.91	71.20	53.75	72.47	47.27	75.43	43.40	76.71	61.67

Table 12: Full perplexity results for Llama-3 70B

Bits (W/A/KV)	Method	WikiText2↓	PTB↓	C4↓	Avg↓
4-4-16	QuaRot	8.10	16.21	15.47	13.26
	DFRot	7.25	14.26	13.60	11.70
	SpinQuant	5.45	11.50	10.98	9.31
	DartQuant	5.04	10.18	9.72	8.31
	Ours f.p. calib.	4.95	10.19	9.60	8.25
	Ours quant. calib.	5.09	10.13	9.65	8.29
4-4-4	QuaRot	8.25	16.61	15.74	13.53
	DFRot	7.40	14.70	13.95	12.02
	SpinQuant	5.49	11.11	10.59	9.06
	DartQuant	5.13	10.35	9.91	8.46
	Ours f.p. calib.	5.04	10.39	9.76	8.40
	Ours quant. calib.	5.18	10.35	9.93	8.49

Table 13: Full common sense reasoning results for Llama-3 70B

Bits (W/A/KV)	Method	WG	SIQA	LAMB	MMLU	ARC-E	ARC-C	HS	OBQA	PIQA	Avg
4-4-16	QuaRot	65.04	43.60	62.82	45.80	69.02	43.26	70.57	40.60	77.09	57.53
	DFRot	68.75	43.3	65.11	50.77	73.7	49.91	75.28	42.6	77.09	60.72
	SpinQuant	76.72	46.26	75.18	63.23	77.82	54.10	80.85	45.80	80.74	66.74
	DartQuant	78.37	46.37	76.29	67.13	79.25	55.38	83.05	46.00	81.56	68.16
	Ours f.p. calib.	76.48	47.13	76.98	67.66	79.97	54.44	81.70	47.60	81.56	68.17
	Ours quant. calib.	76.48	47.85	75.70	66.09	78.28	54.86	82.39	45.00	81.99	67.63
4-4-4	QuaRot	69.93	44.22	61.73	45.31	68.60	42.83	70.06	39.40	76.22	57.59
	DFRot	68.82	42.89	64.53	47.30	71.72	48.63	74.37	41.20	76.88	59.59
	SpinQuant	73.16	45.45	73.37	62.97	77.27	52.82	80.36	43.80	80.52	65.52
	DartQuant	76.87	46.16	75.65	66.42	79.29	54.52	83.03	44.20	82.10	67.58
	Ours f.p. calib.	76.56	46.26	76.17	66.39	79.38	56.40	81.60	45.20	82.15	67.79
	Ours quant. calib.	77.43	48.21	75.41	66.99	79.04	56.91	82.17	45.40	81.61	68.13

F Results on additional architectures

Table 14: Average perplexity and zero-shot common sense reasoning results on additional architectures.

Model	Method	Avg. PPL↓	Zero-shot↑
Mistral 7B	DartQuant	18.60	63.56
	Ours	18.50	63.67
Qwen 3 8B Base	DartQuant	12.53	62.91
	Ours	12.46	63.42

In this section, we present additional results on non-Llama architectures. We evaluate both our proposed method and DartQuant on Mistral 7B [45] and Qwen 3 8B Base [46] and present results in Table 14. Our method achieves both the lowest perplexity and best common sense reasoning performance on both architectures.

G Effect of initialization

Table 15

Initialization	Init. Quant. Error ↓	Final Quant. Error ↓	Avg. PPL↓
Identity	0.1694	0.0127	10.65
Gaussian QR	0.0158	0.0127	10.70
Hadamard	0.0156	0.0127	10.62

As the target hypercube corner is determined by the sign pattern induced by the current rotation, different initializations can produce substantially different initial target assignments and initial quantization errors. We therefore evaluated the sensitivity of our method to three initialization strategies on Llama-3 8B: the identity matrix, a random orthogonal matrix obtained by QR decomposition of a Gaussian matrix, and a Hadamard rotation. We present these results in Table 15. As can be seen, hadamard achieves the lowest initial quantization error. However, all methods achieve nearly the same final quantization error. In addition, average perplexity is similar regardless of initialization.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have claimed a new approach to optimizing rotation matrices for activation quantization. We provide a thorough explanation of our method in Section 3 and experimental validation in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in the "Conclusion and limitations" section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Our primary theoretical contribution is briefly summarized in 3 with the full proof provided in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The method is described in full such that it is reproducible. Calibration settings and quantization settings are described in the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All benchmarks/models are publicly available. Source code is provided as a public github repository.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: To the best of our knowledge, all hyperparameters are provided in the text as well as in code.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the size of experiments and limited access to compute, we are only able to provide results for a single run.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Calibration time is provided in the experiments section and the appendix provides additional hardware details for all experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the Code of Ethics and have adhered to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss broader impacts in our conclusion.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: As the models and benchmarks used are already open source, this paper poses no such risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all code and datasets used and respect the terms of their licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The source code is provided alongside detailed documentation to reproduce the results.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This research did not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were only used for writing, editing, and formatting purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.