Gradual Binary Search and Dimension Expansion : A general method for activation quantization in LLMs

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large language models (LLMs) have become pivotal in artificial intelligence, demonstrating strong capabilities in reasoning, understanding, and generating data. However, their deployment on edge devices is hindered by their substantial size, often reaching several billion parameters. Quantization is a widely used method to reduce memory usage and inference time, however LLMs present unique challenges due to the prevalence of outliers in their activations. In this work, we leverage the theoretical advantages of Hadamard matrices over random rotation matrices to push the boundaries of quantization in LLMs. We demonstrate that Hadamard matrices are more effective in reducing outliers, which are a significant obstacle in achieving low-bit quantization. Our method based on a gradual binary search enables 3-bit quantization for weights, activations, and key-value (KV) caches, resulting in a 40% increase in accuracy on common benchmarks compared to SoTA methods. We extend the use of rotation matrices to support non-power-of-2 embedding dimensions, similar to the Qwen architecture, by employing the Paley's algorithm. Our experimental results on multiple models family like Mistral, LLaMA, and Qwen demonstrate the effectiveness of our approach, outperforming existing methods and enabling practical 3-bit quantization.

1 Introduction

Large Language Models (LLMs) have become a central component of artificial intelligence due to their strong capabilities in reasoning, understanding, and generating data. These impressive capabilities are attributed to the quality of the data used during training, the model architecture, and the size of the model, which often reaches several billion parameters. This size limitation restricts their deployment on edge devices. Quantization is a widely used method to reduce memory usage and inference time [12, 13], but the challenges differ compared to those faced with Convolutional Neural Networks (CNNs) [9, 29].

Weights are relatively easy to quantize for both CNNs and LLMs and can often achieve ternary quantization without significant loss of accuracy [20, 32]. However, activations behave differently in transformer architectures [23]. The presence of outliers in activations makes conventional quantization (symmetric uniform) very challenging, hindering our ability to achieve 4-bit quantization. LLMs are known to produce spikes in its layers and for some tokens that can be handled separately or diffused in the tensor [7, 29].

One very promising approach to overcome this limitation is to use rotation matrices to redistribute weights and activation values, thereby minimizing the impact of outliers [19, 2]. Additionally, methods such as prefix tokens have shown very interesting results in managing outliers in LLMs [4, 24].

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

In this work, we leverage results on rotation matrices to push the boundaries further and enable 3-bit Weights, Activations, KV cache (WAKV) quantization by employing a binary search. We extend this method to a more general approach capable of handling non-power-of-2 embedding dimensions, similar to Qwen. Our main contributions are:

- A theoretical demonstration that Hadamard matrices are more effective in reducing outliers than rotation matrices drawn on the unit sphere.
- 3-bit quantization for weights, activations, and KV cache, resulting in a 40% increase in accuracy on common benchmarks using a gradual binary search.
- Extension of rotation matrices to support non-power-of-2 embedding dimensions using the Paley's algorithm.
- The introduction of dimension expansion to build a more general rotation pipeline allowing architectures like Qwen to work with rotations.

2 Related Works

2.1 Quantization

Quantizing models involves reducing the number of bits required to store and compute model activations. This process is crucial for deploying LLMs on resource-constrained devices. To achieve this, we define a scaling factor that determines the distance between quantization bins and the range of values to be compressed.

For symmetric uniform quantization, we apply a rounding function to a scaled distribution:

$$\hat{X} = \text{round}\left(\frac{X}{\Delta}\right)\Delta,\, \Delta = \frac{\max|X|}{2^b - 1}$$

where Δ is the scaling factor, b is the bitwidth, and $\max|X|$ is the maximum absolute value of the distribution, preserving extreme values for activations.

Such quantization can be applied per-token, where each token has a different scaling factor, or per-tensor, where a single scaling factor is used for each activation tensor [12, 13]. Per-token quantization is more challenging to implement efficiently in practice compared to per-tensor quantization but results in better quantization performances. Scaling factors can be static during inference, based on statistics computed on a subset of the dataset, or dynamic, recomputed at each step.

Quantization can lead to a significant drop in performance when applied post-training (PTQ) [30]. To mitigate this, some methods adapt weights to the noise introduced during a training phase (QAT) [18, 8]. Typically, for LLMs, only linear layers are quantized, as they account for most of the computation cost, while normalization layers, matrix multiplications, and the softmax function within the attention block are left unquantized.

2.2 Outliers

Quantizing LLM weights is relatively straightforward and does not require extensive efforts to achieve. Techniques like GPTQ [11] enables 8-bit quantization without retraining, preserving model accuracy. Some QAT methods can even push the boundaries to 1-bit quantization, as seen in BitNet [26] or ternary quantization [20].

However, LLMs present unique challenges due to the prevalence of extreme high values in their activations [27, 23, 15, 17]. The scaling factor, which is directly tied to the maximum absolute value, often causes most of the distribution to be rounded to zero, leading to performance degradation. To address this, techniques like LLM.int8() [7] cluster these outliers and quantize them separately from the main distribution.

Alternative methods, such as SmoothQuant [29], shift the quantization challenge from activations to weights by introducing a scaling parameter between them. Other approaches attempt to relocate these spikes into "sink tokens" before quantization [24]. Some research focuses on understanding the upstream causes of these spikes during the learning process to limit their impact post-training [23].

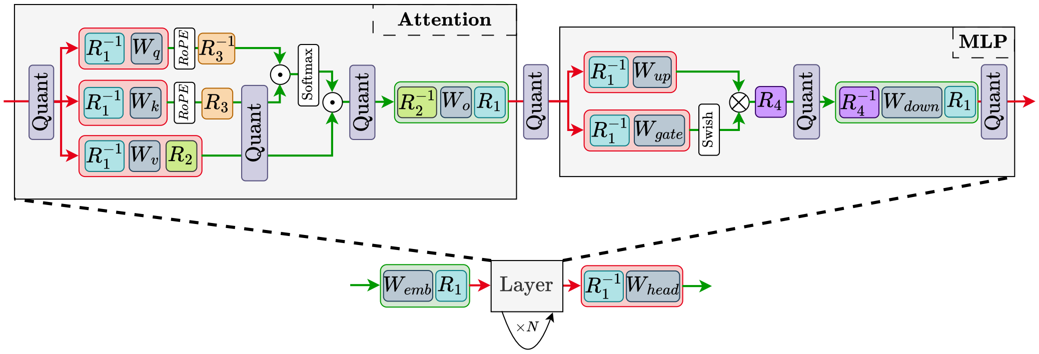


Figure 1: Architecture's pipeline with rotation matrices R_1, R_2, R_3, R_4 and dimension expansion. Red lines represents expanded tokens in $4096+d$ dimensions and green lines represents non expanded tokens. Projections in red (QKV, Gate, Up and LM_{head}) have their input weigths dimension expanded and projections in green (Out, Down and Embeddings) have their output weights dimension expanded

Additionally, efforts are made to better locate these outliers by visualizing the layers, dimensions, and tokens that may be their source [22].

2.3 Rotation Matrices

2.3.1 Random orthogonal matrices

Rotation matrices play a pivotal role in various applications, including signal processing, computer vision, and machine learning. These matrices are orthogonal and invertible by their transpose, meaning they preserve the length of vectors and the angles between them. In the context of quantization, rotation matrices can be used to decorrelate and redistribute the energy of model activations [1, 2, 3], making them more amenable to quantization. The idea is to apply orthogonal matrices before quantization to flatten the distribution and then recover the tensor by applying its inverse (see Figure 1). Part of this process can be pre-computed and fused with weights and the rest needs to be done at inference [1].

However, the effectiveness of rotation matrices depends on the specific matrix used. Randomly drawn orthogonal rotation matrices can introduce noise and reduce the overall performance of the model. To mitigate this, some methods adapt the rotation matrices during a training phase to better align with the model's weights and activations [19].

In practice, rotation matrices are often used in conjunction with other quantization techniques, such as GPTQ. This combination allows for more robust and efficient quantization of large language models, enabling their deployment on resource-constrained devices.

2.3.2 Hadamard matrices

Hadamard matrices are another powerful tool in the quantization arsenal. These matrices are orthogonal matrices and all their entries are either +1 or -1 making them very efficient to compute (eq 1). Hadamard matrices have been extensively used in signal processing, error-correcting codes [14], and more recently, in the quantization of neural networks [1, 2].

One of the key advantages of Hadamard matrices is their ability to decorrelate the activations of a model. By applying a Hadamard matrix, the activations are transformed into a new basis where the correlations between different dimensions are minimized. This decorrelation property is particularly useful in reducing the impact of outliers, as the extreme values are spread out across multiple dimensions.

Hadamard matrices of order 2^n can be constructed recursively using the Fast Hadamard Transform (FHT) method: For $n \geq 1$, construct the $2^{n+1} \times 2^{n+1}$ Hadamard matrix $H_{2^{n+1}}$ using the $2^n \times 2^n$

Hadamard matrix H_{2^n} as follows:

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad H_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}, \quad H_{2^{n+1}} = \begin{pmatrix} H_{2^n} & H_{2^n} \\ H_{2^n} & -H_{2^n} \end{pmatrix} \tag{1}$$

This method is highly efficient for generating Hadamard matrices and can be applied in real-time. In summary, both rotation matrices and Hadamard matrices are essential for the quantization of large language models. However, Hadamard matrices offer several advantages: they can be generated more efficiently, their structure of containing only 1 and -1 makes them highly efficient for matrix multiplication, and they are known to handle outliers in activations more effectively [19]. In the following sections, we will theoretically demonstrate that Hadamard matrices are more effective than random rotation matrices drawn from the unit sphere in reducing the amplitude of outliers.

2.3.3 Paley algorithm

To generate other dimensions n for Hadamard matrix we can use known small matrices and apply power of 2 algorithm as used in QuaRot [2] but it can be limiting and doesn't cover a lot of values. To overcome this issue we can use the Paley's Algorithm that generate a Hadamard matrix $n \times n$ if $n-1$ is a prime number and $n-1 \equiv 3 \pmod 4$. This algorithm is described below (Algorithm 2) and needs to generate Legendre symbols $\left(\frac{a}{p}\right)$ which take any integer number a and prime number p to produce a value in $\{-1, 0, 1\}$ as below :

- If a is a quadratic residue modulo p, then there exists an integer x such that $x^2 \equiv a \pmod p$. In this case, $\left(\frac{a}{p}\right) = 1$.
- If a is a quadratic non-residue modulo p, then there is no integer x such that $x^2 \equiv a \pmod p$. In this case, $\left(\frac{a}{p}\right) = -1$.
- If $a \equiv 0 \pmod p$, then $\left(\frac{a}{p}\right) = 0$.

Generating Legendre symbols can be time-consuming, especially for high-dimensional matrices. However, in the following sections, we will use this algorithm to generate non-power-of-2 Hadamard matrices and fuse them with the weights, so we only need to compute the Legendre symbols once.

3 Analysis and theoretical demonstrations

3.1 Clipping Ratio

To perform quantization we can play on several parameters to improve the effectiveness of the process, for example in LSQ [9] they optimise the scaling factor trough training, or FracBits [31] which tries to find the best precision for every layer. Other works highlighted the importance of the clipping ratio like PACT [5] where the optimization is done during training. Some others apply a Grid Search [4] to find the best configuration particularly useful for LLMs where training or fine tuning can be very time consuming.

Clipping ratios are essential for managing outliers, as they establish the balance between maintaining high precision for small values and preserving a maximum value close to its original. However, the model exhibits significant variability in how quantization responds to changes in the clipping ratio. While some projections can tolerate very low clipping ratios, others experience a substantial accuracy drop with even slight adjustments (see Appendix C). Therefore, to effectively manage this variability, a tailored clipping ratio must be determined for each projection.

Previous studies have shown that quantization error is not always the best metric to guide the optimization process for quantization parameters [21]. Specifically, at very low precision levels, such as 4 or 3 bits, the set of quantized weights deviates significantly from the optimized configuration obtained during training. Attempting to recover this configuration using quantization error often results in an ineffective set of weights. To address this issue, we can use perplexity as an objective function.

Perplexity provides a more accurate representation of model performance and is computationally efficient, as it is based on Cross Entropy Loss, which is frequently used during training for its smoothness.

3.2 Hadamard Matrices reduce outliers more

Experimentally, it is observed that Hadamard matrices tend to reduce better the amplitude of outliers present in the layers of LLMs, which directly impacts the performance of these models. However, the question of why such a phenomenon occurs has remained open from a theoretical perspective. We now provide an answer to this question.

Theorem 3.1 (Hadamard reduction)**.** *$\forall x \in \mathbb{R}^n$ containing an outlier, i.e., $x = (c, \epsilon, ..., \epsilon)^T$ with $c >> \epsilon$ we have*

$$\max_{1\leq i\leq n} |(Hx)_i| \leq \max_{1\leq i\leq n} |(Qx)_i|$$

with H a Hadamard matrix belonging to $\mathbb{R}^{n\times n}$ and Q a rotation matrix drawn randomly on the unit sphere $\mathcal{S}^{n-1} = \{x \in \mathbb{R}^n : ||x||_2 = 1\}$.

To demonstrate Theorem 3.1, we can calculate the two terms of the inequality and thus show the superiority of one over the other.

Lemma 3.1 (Hadamard incoherence)**.** *For H a Hadamard matrix belonging to $\mathbb{R}^{n\times n}$ and $x = (c, \epsilon, ..., \epsilon)^T$ with $c >> \epsilon$*

$$\max_{1\leq i\leq n} |(Hx)_i| = \frac{c}{\sqrt{n}}$$

Lemma 3.2 (Rotation incoherence)**.** *For Q a rotation matrix drawn randomly on the unit sphere $\mathcal{S}^{n-1} = \{x \in \mathbb{R}^n : ||x||_2 = 1\}$ and $x = (c, 0, ..., 0)^T$ with $c >> 1$*

$$\max_{1\leq i\leq n} |(Qx)_i| = c\sqrt{\frac{2\log n}{n}}$$

We can prove in Lemma 3.1 and Lemma 3.2 that the reduction of outliers with a Hadamard matrix is of order $O(\frac{1}{\sqrt{n}})$ and $O(\sqrt{\frac{2\log n}{n}})$ for a random orthogonal matrix (demonstrations are done in Appendix A). These results prove Theorem 3.1 and also show the close link between reduction and the dimension of embeddings in LLMs. The higher the dimension is the stronger the reduction will be. We can also demonsatrate in Theorem 3.2 that Hadamard matrices are optimal and the best group of matrices to reduce outliers.

Theorem 3.2 (Hadamard optimality)**.** *For any orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ we have:*

$$\max_{1\leq i,j\leq n} |Q_{ij}| \geq \frac{1}{\sqrt{n}}$$

And a Hadamard matrix reaches this bound.

4 Method

4.1 Gradual Binary Search

In Section 3.1, we emphasize the importance of the clipping ratio parameter and its significant impact on model performance. We stress the need to optimize each projection with its own clipping ratio for best results. Our primary contribution is an algorithm that determines the optimal clipping ratio for each quantizer using a binary search (Algorithm 1). To drive the binary search, we minimize perplexity across various clipping ratios, assuming a single minimum and a convex landscape. Additionally, we quantize our model gradually: first, we quantize and optimize the initial linear projection while keeping the rest in FP16, then use the obtained parameters to quantize and optimize the next projection, and so on. This process is discussed in Appendix C where we experimentally show the necessity to optimize gradually the clipping ratio.

Algorithm 1 Gradual Binary Search

```
Require: A model $M$, a dataset $D$, a threshold $\epsilon$
Ensure: A list L of clipping ratios                    ▷ + operand on L means concatenation
n ← number of projections in M
$L \leftarrow []$
for $i \leftarrow 1$ to $n-1$ do
    $a \leftarrow 0$
    $b \leftarrow 1$
    $m \leftarrow (a+b)/2$                             ▷ We keep track of the middle element
    $M \leftarrow$ quantize_proj$(M, i)$               ▷ Quantize projection $i$ of model $M$
    $f_m =$ evaluate$(M, D, L + m)$                    ▷ Evaluate model $M$ on dataset $D$ with clipping ratios L
    iteration ← 0
    while $b - a > \epsilon$ do                        ▷ We iterate until we converge
        if iteration is even then                     ▷ Allows to use only one loop for binary search
            $x \leftarrow (a+m)/2$
        else
            $x \leftarrow (b+m)/2$
        end if
        $f_x \leftarrow$ evaluate$(M, D, L + x)$       ▷ Evaluate model on a new clipping ratio
        if $f_x < f_m$ then                            ▷ If we improve PPL (the lower the better) we keep it
            if $x < m$ then                            ▷ If the target is less than the middle, search the left half
                $b \leftarrow m$
            else                                      ▷ If not , search the right half
                $a \leftarrow m$
            end if
            $m, f_m \leftarrow x, f_x$
        else
            if $x < m$ then
                $a \leftarrow x$
            else
                $b \leftarrow x$
            end if
        end if
        iteration ← iteration + 1
    end while
    $L = L + m$                                       ▷ Add new element to the list
end for
return $L$
```

4.2 Increasing dimensions

Lemma 4.1 (Expanding limit)**.** *For a matrix product AB with $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$ in b bits and $A'B'$ with $A' \in \mathbb{R}^{m \times (n+d)}$ and $B' \in \mathbb{R}^{(n+d) \times p}$ in b' bits we must have $d \leq \frac{n(b-b')}{b'}$ so that BitOps$(A'B') \leq$ BitOps(AB), with $m, n, p, b, b' \in \mathbb{N}$ and $b' \leq b$*

One important limitation of QuaRot's implementation of rotation matrices in LLMs is the necessity to have embeddings in a power of 2 dimension which can be very limiting in some architectures like Qwen2.5-1.5B which have features in dimension 1536 and can not be quantized with QuaRot. To overcome this problem we increase manually the dimension of embedding by adding zeros in the weights (independently developed in [10]) to reach a dimension suitable to generate a Hadamard matrix with the Paley's algorithm 2. Then we save the matrix product of weights padded with 0s and the Hadamard matrix as our new weights (see figure 1 and Eq 2 for an example in dimension 4). The primary goal is to create a more versatile pipeline compatible with any architecture but it also enhance performance through increased dimensionality.

$$W \leftarrow \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \times \begin{pmatrix} a & b \\ c & d \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{2}$$

Indeed theorem 3.1 ensures that increasing the dimension helps reduce the impact of outliers in any tensor. Consequently, by adding zeros to the weight tensors, we also improve the effectiveness of

Table 1: Results in 4 bits WAKV quantization on perplexity (PPL), PIQA, hellaswag (HS), arc-easy (ARC-E), arc-challenge (ARC-C), winogrande (WINO) and lambada, we also compute the average value (AVG) which represents a % of success. We compare our method, GBS, with QuaRot (where QuaRot^{+} indicates the use of dimension expansion) and clearly observe that GBS outperforms QuaRot across all computed metrics. Additionally, dimension expansion enables QuaRot to be compatible with Qwen's family of models.

Model	Method	PPL↓	PIQA	HS	ARC-E	ARC-C	Wino	Lambada	AVG↑
Mistral 7B Inst v0.3	FP16	5.49	71.27	74.6	67.94	74.27	58.96	82.66	71.62
	QuaRot	5.98	67.7	71.45	63.94	69.53	55.2	79.46	67.88
	QuaRot + GBS	**5.75**	**70.55**	**74.44**	**66.66**	**71.82**	**56.66**	**80.77**	**70.15**
Mistral 7B v0.1	FP16	5.25	72.49	75.59	69.4	73.95	54.86	80.18	71.08
	QuaRot	5.82	67.62	71.88	63.36	70.01	48.98	76.6	66.41
	QuaRot + GBS	**5.57**	**71.47**	**74.95**	**68**	**72.22**	**51.54**	**79.21**	**69.56**
Llama2 7B	FP16	5.47	71.08	73.9	68.25	68.98	46.33	74.58	67.19
	QuaRot	6.21	64.65	69.09	60.22	64.64	**43.17**	69.78	61.92
	QuaRot + GBS	**6.04**	**65.64**	**69.88**	**61.4**	**66.46**	42.32	**70.75**	**62.74**
Llama3 8B	FP16	6.13	72.62	76.01	69.22	72.93	53.41	77.69	70.3
	QuaRot	8.33	61.66	66.27	57.05	64.72	42.06	68.06	59.97
	QuaRot + GBS	**7.4**	**67.87**	**72.02**	**63.73**	**71.03**	**45.9**	**73.7**	**65.71**
Qwen2.5 7B Inst	FP16	7.45	66.23	69.73	62.74	70.56	55.12	81.02	67.57
	QuaRot	-	-	-	-	-	-	-	-
	QuaRot^{+}	9.21	56.66	59.31	54.01	63.54	48.89	69.78	58.7
	QuaRot^{+} + GBS	**8.23**	**62.58**	**64.91**	**60.24**	**66.61**	**49.4**	**72.39**	**62.69**
Qwen2.5 1.5B Inst	FP16	9.64	58.09	61.21	54.98	63.3	46.59	75.8	60.0
	QuaRot	-	-	-	-	-	-	-	-
	QuaRot^{+}	14.44	39.05	40.23	37.86	54.85	35.75	58.71	44.41
	QuaRot^{+} + GBS	**12.05**	**43.94**	**45.24**	**42.64**	**58.64**	**39.33**	**65.61**	**49.23**

quantization. The intuition behind it is that by adding more dimensions in our tensors we create more space to store information and especially outliers which will be sliced in more parts and recovered better after quantization. This process increases the model size and computational cost, necessitating a trade-off to achieve better accuracy without a significant increase in computational requirements.

Since we only expand the input and output dimensions of the attention block and Multi-Layer Perceptron (MLP), there is no additional computational cost at inference. The new weights are stored, and the Hadamard matrices required for inference remain unchanged, allowing them to be efficiently computed using the FHT.

Lemma 4.1 shows the threshold after which the increase in dimentionality is worse than just quantizing with one more bit. For example with a LLaMA3-8B which has embeddings in 4096 dimensions we are only allowed to increase to $d = 1366$ dimensions in 3 bits before reaching the computational cost in 4 bits.

5 Experiments

5.1 Setup

We conduct our experiment based on the the code of QuaRot which performs per-token quantization for activations and GPTQ for weights. We also quantize KV caches using asymmetric quantization with a group size of 128. We compare our results on several metrics : perplexity (PPL) on WikiText2, and 6 benchmarks : PIQA, hellaswag (HS), arc-easy (ARC-E), arc-challenge (ARC-C), winogrande (WINO) and lambada, we also compute the average value (AVG) of these 6 benchmarks. We performs ours experiments in 4 and 3 bits quantization on 6 different models from the Mistral library, LLaMA architecture and Qwen. We used only one GPU A100 to perform quantization and Gradual Binary Search (GBS) with 10% of the train set of WikiText2 for 4 days for the biggest models.

Table 2: Results in 3 bits WAKV quantization on perplexity (PPL), PIQA, hellaswag (HS), arc-easy (ARC-E), arc-challenge (ARC-C), winogrande (WINO) and lambada, we also compute the average value (AVG) which represents a % of success. We compare our method, GBS, with QuaRot (where QuaRot$^+$ indicates the use of dimension expansion) and clearly observe that GBS outperforms QuaRot across all computed metrics. Additionally, dimension expansion enables QuaRot to be compatible with Qwen's family of models.

Model	Method	PPL↓	PIQA	HS	ARC-E	ARC-C	Wino	Lambada	AVG↑
Mistral 7B Inst v0.3	FP16	5.49	71.27	74.6	67.94	74.27	58.96	82.66	71.62
	QuaRot	38.28	7.66	9.49	5.82	51.46	23.55	34.39	22.06
	QuaRot + GBS	**7.04**	**62.17**	**66.93**	**57.4**	**61.56**	**46.42**	**73.44**	**61.32**
Mistral 7B v0.1	FP16	5.25	72.49	75.59	69.4	73.95	54.86	80.18	71.08
	QuaRot	100.85	3.07	4.13	2.0	48.62	22.18	30.6	18.43
	QuaRot + GBS	**7.31**	**59.22**	**64.41**	**54.03**	**63.3**	**40.53**	**68.39**	**58.31**
Llama2 7B	FP16	5.47	71.08	73.9	68.25	68.98	46.33	74.58	67.19
	QuaRot	332.56	0.25	0.47	0.04	51.14	26.11	30.39	18.07
	QuaRot + GBS	**9.18**	**39.03**	**49.91**	**28.14**	**56.12**	**31.91**	**54.92**	**43.34**
Llama3 8B	FP16	6.13	72.62	76.01	69.22	72.93	53.41	77.69	70.3
	QuaRot	1315	0.05	0.08	0.02	49.41	23.72	27.78	16.84
	QuaRot + GBS	**12.62**	**44.92**	**50.32**	**39.51**	**60.22**	**32.85**	**53.7**	**46.92**
Qwen2.5 7B Inst	FP16	7.45	66.23	69.73	62.74	70.56	55.12	81.02	67.57
	QuaRot	-	-	-	-	-	-	-	-
	QuaRot$^+$	251	1.14	1.14	1.14	49.33	25.0	32.15	18.32
	QuaRot$^+$ + GBS	**12.33**	**41.9**	**42.62**	**41.18**	**56.59**	**40.53**	**61.83**	**47.44**
Qwen2.5 1.5B Inst	FP16	9.64	58.09	61.21	54.98	63.3	46.59	75.8	60.0
	QuaRot	-	-	-	-	-	-	-	-
	QuaRot$^+$	3411	0.06	0.12	0.0	49.72	**23.98**	27.99	16.98
	QuaRot$^+$ + GBS	**34.97**	**13.84**	**14.81**	**12.87**	**52.17**	23.72	**38.89**	**26.05**

5.2 Results

5.2.1 Gradual Binary Search performances

Table 1 and Table 2 shows the results in 4 and 3 bits quantization on the perplexity and 6 benchmarks. In 4 bits our method GBS clearly outperforms previous methods for all models improving up to almost 6% for LLaMA3-8B, 5% on Qwen2.5 1.5B Instruct, 4% on Mistral 7B and 3% on Mistral 7B Instruct.

In 3 bits GBS made activation quantization possible with an increase of accuracy reaching 40% for Mistral 7B (Table 2). All other models have been greatly affected by GBS reducing the gap with 4 bits quantization. PPL is also significantly impacted by GBS reducing by a factor of 100 in the case of LLaMA3-8B. We now have a method reaching decent performances in 3 bits like with Mistral 7B Instruct which is only 10% less than FP16 and reach 61.32% accuracy on our benchmarks.

GBS appears to be highly effective in enhancing quantization performance, supporting our hypothesis that optimizing Perplexity via binary search is preferable to minimizing quantization error. We assumed a single minimum and a convex function, allowing us to leverage binary search while relying on the smoothness of CrossEntropy—an assumption that appears to hold true. Perplexity emerges as a strong objective for guiding our optimization, as it correlates well with improved benchmark performance (see Appendix D for more details).

We also evaluated GBS using two additional methods: SpinQuant [19] and DFRot [28], both relying on rotation matrices (see Appendix E). Our binary search approach, particularly when restricted to 3 bits, significantly enhances their performance, thus validating the broad effectiveness of our method

5.2.2 Matrix expansion effect

To execute QuaRot on Qwen2.5 1.5B, as shown in Tables 1 and 2, we needed to increase the embedding dimension (as detailed in Section 4.2) to a value that can generate a Hadamard matrix. Initially, 1536 was not suitable since it is neither a power of 2 nor divisible by a known dimensions like 172, 156, or 140. We first chose to add 8 dimensions, reaching 1542, which can be managed by the Paley algorithm without excessive computational costs. This algorithm is also used for Qwen2.5

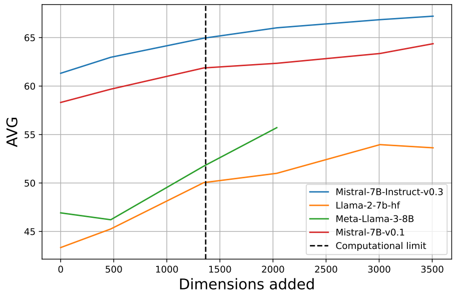


Figure 2: Effect of expanding dimensions on 6 benchmarks average (AVG) for different models in 3 bits WAKV quantization and the computational limit of Lemma 4.1. Due to memory constraints on GPU A100 we could not increase more than 2036 dimensions for LLaMA3-8B.

7B to generate a Hadamard matrix of dimension 3584 and expand MLP to 32 dimensions, enabling the Qwen architecture to work with QuaRot. This process is applicable to any architecture and any embedding dimension without increasing computational costs making this method very general.

We now study the impact of expanding dimensions on performance. Figure 2 show the evolution of AVG with the number of dimensions added to our tokens and we clearly see the positive impact on performances. We can reach 68.95% of accuracy for Mistral-7B Instruct but at a very high computational cost.

Another beneficial aspect of dimension expansion is seen in Group Local Rotation, introduced in QuaRot and explored in [16]. This technique involves decomposing a tensor into smaller sub-tensors and applying the same small power-of-2 Hadamard matrix to each of these sub-tensors. This approach leverages efficient Hadamard transforms (as introduced in Section 2.3.2) and significantly speeds up inference. Particularly for MLP layers that often operate in high-dimensional spaces, expanding dimensions can help identify a more suitable divisor, resulting in efficient power-of-2 sub-tensors.

6 Conclusion

In this work, we introduced an approach to optimize the quantization of LLMs using Gradual Binary Search and Hadamard matrices. Our method achieves efficient 3-bit quantization for weights, activations, and key-value caches, significantly improving model performance. We also theoretically demonstrated that Hadamard matrices are more effective than random rotation matrices in reducing extreme values in activations.

We also extended the use of rotation matrices to support non-power-of-2 embedding dimensions using the Paley algorithm and dimension expansion. This generalization allows our method to be applied to various architectures, including those with unique embedding dimensions. Experimental results on models from the Mistral library, LLaMA architecture, and Qwen show the effectiveness of our approach, outperforming existing methods.

Overall, our findings suggest that GBS and Hadamard matrices have great potential for advancing LLM quantization, making them more suitable for resource-constrained devices. Future work will explore mix computation and combining GBS with other methods.

7 Limitations

As explained in the previous part expanding dimensions has a big computational cost and it worsen with context length that is why we need to be aware of the expanding limit. One potential solution is to implement a Mixed Computation pipeline, where dimensions are only expanded in specific layers based on the presence of outliers, thereby substantially reducing computational overhead.

Another challenge arises with GBS, which involves computing perplexity at each step—a process that can be time-consuming for PTQ methods, sometimes taking several days. To mitigate this, we could use less than 10% of WikiText2 which might be excessive and unnecessary.

References

[1] Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress Large Language Models by Deleting Rows and Columns, February 2024. URL http://arxiv.org/abs/2401.15024. arXiv:2401.15024.

[2] 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, October 2024. URL http://arxiv.org/abs/2404.00456. arXiv:2404.00456 [cs].

[3] Jerry Chee, Volodymyr Kuleshov, and Yaohui Cai. QuIP: 2-Bit Quantization of Large Language Models With Guarantees.

[4] Mengzhao Chen, Yi Liu, Jiahao Wang, Yi Bin, Wenqi Shao, and Ping Luo. PrefixQuant: Static Quantization Beats Dynamic through Prefixed Outliers in LLMs, October 2024. URL http://arxiv.org/abs/2410.05265. arXiv:2410.05265.

[5] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I.-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. PACT: Parameterized Clipping Activation for Quantized Neural Networks, July 2018. URL http://arxiv.org/abs/1805.06085. arXiv:1805.06085 [cs].

[6] Laurens De Haan and Ana Ferreira. *Extreme Value Theory*. Springer Series in Operations Research and Financial Engineering. Springer, New York, NY, 2006. ISBN 978-0-387-23946-0 978-0-387-34471-3. doi: 10.1007/0-387-34471-3. URL http://link.springer.com/10.1007/0-387-34471-3.

[7] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale.

[8] Alexandre Défossez, Yossi Adi, and Gabriel Synnaeve. Differentiable Model Compression via Pseudo Quantization Noise, October 2022. URL http://arxiv.org/abs/2104.09987. arXiv:2104.09987 [cs, stat].

[9] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned Step Size Quantization, May 2020. URL http://arxiv.org/abs/1902.08153. arXiv:1902.08153 [cs, stat].

[10] Giuseppe Franco, Pablo Monteagudo-Lago, Ian Colbert, Nicholas Fraser, and Michaela Blott. Improving Quantization with Post-Training Model Expansion, March 2025. URL http://arxiv.org/abs/2503.17513. arXiv:2503.17513 [cs].

[11] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers, March 2023. URL http://arxiv.org/abs/2210.17323. arXiv:2210.17323 [cs].

[12] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. A Survey of Quantization Methods for Efficient Neural Network Inference, June 2021. URL http://arxiv.org/abs/2103.13630. arXiv:2103.13630 [cs].

[13] Yunhui Guo. A Survey on Methods and Theories of Quantized Neural Networks, December 2018. URL http://arxiv.org/abs/1808.04752. arXiv:1808.04752 [cs, stat].

[14] K. J. Horadam. *Hadamard Matrices and Their Applications*. Princeton University Press, January 2012. ISBN 978-1-4008-4290-2. Google-Books-ID: oR_HDgAAQBAJ.

[15] Xijie Huang, Zechun Liu, Shih-Yang Liu, and Kwang-Ting Cheng. RoLoRA: Fine-tuning Rotated Outlier-free LLMs for Effective Weight-Activation Quantization, July 2024. URL http://arxiv.org/abs/2407.08044. arXiv:2407.08044 [cs].

[16] Sangjin Kim, Yuseon Choi, Jungjun Oh, Byeongcheol Kim, and Hoi-Jun Yoo. LightRot: A Light-weighted Rotation Scheme and Architecture for Accurate Low-bit Large Language Model Inference. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, pages 1–1, 2025. ISSN 2156-3365. doi: 10.1109/JETCAS.2025.3558300. URL `https://ieeexplore.ieee.org/document/10950449/`.

[17] 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.

[18] Ji Lin, Chuang Gan, and Song Han. Defensive Quantization: When Efficiency Meets Robustness, April 2019. URL `http://arxiv.org/abs/1904.08444`. arXiv:1904.08444 [cs, stat].

[19] 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, May 2024. URL `http://arxiv.org/abs/2405.16406`. arXiv:2405.16406 [cs].

[20] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits, February 2024. URL `http://arxiv.org/abs/2402.17764`. arXiv:2402.17764 [cs].

[21] Lucas Maisonnave, Cyril Moineau, Olivier Bichler, and Fabrice Rastello. Applying maximum entropy principle on quantized neural networks correlates with high accuracy.

[22] Lucas Maisonnave, Cyril Moineau, Olivier Bichler, and Fabrice Rastello. Precision Where It Matters: A Novel Spike Aware Mixed-Precision Quantization Strategy for LLaMA-based Language Models, April 2025. URL `http://arxiv.org/abs/2504.21553`. arXiv:2504.21553 [cs].

[23] Aniruddha Nrusimha, Mayank Mishra, Naigang Wang, Dan Alistarh, Rameswar Panda, and Yoon Kim. Mitigating the Impact of Outlier Channels for Language Model Quantization with Activation Regularization, April 2024. URL `http://arxiv.org/abs/2404.03605`. arXiv:2404.03605 [cs] version: 1.

[24] Seungwoo Son, Wonpyo Park, Woohyun Han, Kyuyeun Kim, and Jaeho Lee. Prefixing Attention Sinks can Mitigate Activation Outliers for Large Language Model Quantization. June 2024. URL `https://www.semanticscholar.org/paper/1601ad7616681ed9d7e1b9a04b64c1ad9c7196c7`.

[25] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 2018. ISBN 978-1-108-41519-4. doi: 10.1017/9781108231596. URL `https://www.cambridge.org/core/books/highdimensional-probability/797C466DA29743D2C8213493BD2D2102`.

[26] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. BitNet: Scaling 1-bit Transformers for Large Language Models, October 2023. URL `http://arxiv.org/abs/2310.11453`. arXiv:2310.11453 [cs].

[27] 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 optimal shifting and scaling, October 2023. URL `http://arxiv.org/abs/2304.09145`. arXiv:2304.09145 [cs].

[28] Jingyang Xiang and Sai Qian Zhang. DFRot: Achieving Outlier-Free and Massive Activation-Free for Rotated LLMs with Refined Rotation, December 2024. URL `http://arxiv.org/abs/2412.00648`. arXiv:2412.00648 [cs].

[29] 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 *Proceedings of the 40th International Conference on Machine Learning*, pages 38087–38099.

PMLR, July 2023. URL https://proceedings.mlr.press/v202/xiao23c.html. ISSN: 2640-3498.

[30] Dawei Yang, Ning He, Xing Hu, Zhihang Yuan, Jiangyong Yu, Chen Xu, and Zhe Jiang. Post-Training Quantization for Re-parameterization via Coarse & Fine Weight Splitting, December 2023. URL http://arxiv.org/abs/2312.10588. arXiv:2312.10588 [cs].

[31] Linjie Yang and Qing Jin. FracBits: Mixed Precision Quantization via Fractional Bit-Widths. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10612–10620, May 2021. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v35i12.17269. URL https://ojs.aaai.org/index.php/AAAI/article/view/17269.

[32] Chenzhuo Zhu, Song Han, Huizi Mao, and William J. Dally. Trained Ternary Quantization, February 2017. URL http://arxiv.org/abs/1612.01064. arXiv:1612.01064 [cs].

A Theoretical proofs

A.1 Hadamard matrices

A.1.1 Proof

Proof of Lemma 3.1. We define a Hadamard matrix as a rotation matrix with values equal to 1 or -1 only. By definition, the equality $HH^T = I_n$ must be respected, but without normalization, we have $HH^T = nI_n$. If we decide to normalize this Hadamard matrix by a factor of $\frac{1}{\sqrt{n}}$, we obtain the identity by multiplying it by its transpose. In the case of a vector $x = (c, \epsilon, ..., \epsilon)^T$ with $c >> \epsilon$, applying a Hadamard matrix to it amounts to multiplying the maximum absolute value by $\frac{1}{\sqrt{n}}$ since all the values of H are either $\frac{1}{\sqrt{n}}$ or $-\frac{1}{\sqrt{n}}$, hence the desired result. □

Proof of Lemma 3.2. Let Q be a rotation matrix drawn on the unit sphere $\mathcal{S}^{n-1}$. We assume that the problem is in high dimension, which allows us to approximate the distribution of the elements of the matrix Q:

$$Q_{ij} \sim \mathcal{N}\left(0, \frac{1}{n}\right)$$

This theorem is a classic result of high-dimensional probability theory [25]. It is this result that allows the entire demonstration, because it is from this approximation that we can use the fundamental properties of the extreme values of a normal law. Indeed, for all $i, j \leq n$, $Z_{ij} \sim \mathcal{N}(0, 1)$, then $Q_{ij} = \frac{Z_{ij}}{\sqrt{n}}$. We can show [6] that

$$\mathbb{E}\left[\max_{1\leq i,j\leq n} |Z_{ij}|\right] = \sqrt{2\log n} \tag{3}$$

Therefore,

$$\mathbb{E}\left[\max_{1\leq i,j\leq n} |Q_{ij}|\right] = \sqrt{\frac{2\log n}{n}} \tag{4}$$

Using Talagrand's inequality for a Lipschitz function (Qx is indeed a Lipschitz function) we have:

$$\mathbb{P}\left(\left|\max_{1\leq i,j\leq n} |Q_{ij}| - \sqrt{\frac{2\log n}{n}}\right| > \epsilon\right) \leq 2e^{-Cn\epsilon^2} \tag{5}$$

With $C > 0$. Thus, for sufficiently large n, we have a very high probability of having:

$$\max_{1\leq i,j\leq n} |Q_{ij}| = \sqrt{\frac{2\log n}{n}} \tag{6}$$

We now use this result when applying Q to a vector $x = (c, \epsilon, ..., \epsilon)^T$ with $c >> \epsilon$

$$\max_{1\leq i\leq n} |(Qx)_i| = c \max_{1\leq i,j\leq n} |Q_{ij}| = c\sqrt{\frac{2\log n}{n}}$$

□

Finally, using Lemmas 3.1 and 3.2, we show that for sufficiently large n

$$\max_{1\leq i\leq n} |(Hx)_i| \leq \max_{1\leq i\leq n} |(Qx)_i|$$

We even show that we cannot do better than the Hadamard matrix to redistribute the energy of a matrix.

Proof of Theorem 3.2. Let $Q \in \mathbb{R}^{n\times n}$ be an orthogonal matrix. By definition, for any column i of this matrix:

$$||Q_i||_2 = 1 \Rightarrow \sum_j |Q_{i,j}|^2 = 1$$

However,

$$\sum_j \max_{p,k} |Q_{p,k}|^2 \geq \sum_j |Q_{i,j}|^2 \tag{7}$$

$$\Rightarrow \sum_j \max_{p,k} |Q_{p,k}|^2 \geq 1 \tag{8}$$

$$\Rightarrow n * \max_{p,k} |Q_{p,k}|^2 \geq 1 \tag{9}$$

$$\Rightarrow \max_{p,k} |Q_{p,k}| \geq \frac{1}{\sqrt{n}} \tag{10}$$

We have just shown that an orthogonal matrix cannot reduce an outlier by more than a factor of $\frac{1}{\sqrt{n}}$, and as seen with Lemma 3.1, a Hadamard matrix can reach this bound, it is therefore optimal. □

A.1.2 Experimental Verifications

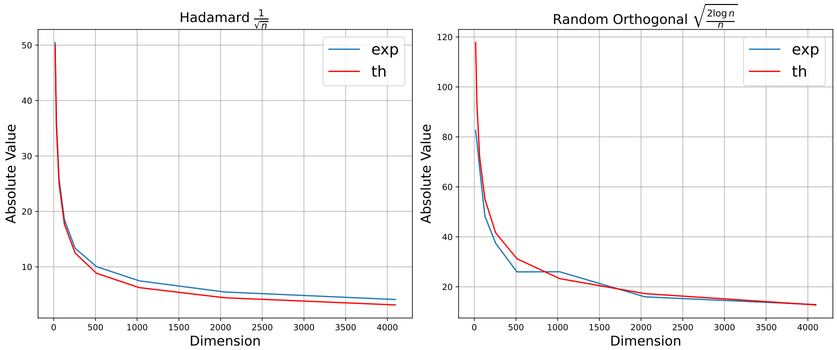


Figure 3: Maximum absolute value as a function of dimension for a randomly drawn rotation matrix and a Hadamard matrix applied to a vector containing a peak at 200 obtained experimentally (blue) and theoretically (red)

To verify the previous theoretical results, we set up a simple experiment where we apply a randomly drawn rotation matrix and a Hadamard matrix to a vector of dimension n following a standard normal distribution with a standard deviation of 0.1 to which we add a peak at 200. We plot the curve of the maximum absolute value after applying our matrices as a function of the dimension n as well as the theoretical curve in Figue 3.

It clearly shows that the theoretical and experimental curves follow each other perfectly, which seems to confirm the previously demonstrated theorems. Hadamard matrices are therefore theoretically and experimentally the most suitable matrices for reducing the impact of an outlier in a vector.

A.2 Increasing dimensions

Proof of Lemma 4.1. We define BitOps as the function that compute the number of operations for a matrix multiplication: BitOps$(AB) = mn^2pb^2$. And we want to find a condition that ensure

$$\begin{aligned}
\text{BitOps}(A'B') &\leq \text{BitOps}(AB) && (11)\\
\Rightarrow \quad m(n+d)^2pb'^2 &\leq mn^2pb^2 && (12)\\
\Rightarrow \quad (n+d)^2b'^2 &\leq n^2b^2 && (13)\\
\Rightarrow \quad (n+d)b' &\leq nb && (14)\\
\Rightarrow \quad nb' + db' &\leq nb && (15)\\
\Rightarrow \quad d &\leq \frac{n(b-b')}{b'} && (16)
\end{aligned}$$

□

B Paley Algorithm

Algorithm 2 Hadamard Matrix Construction using the Paley Method

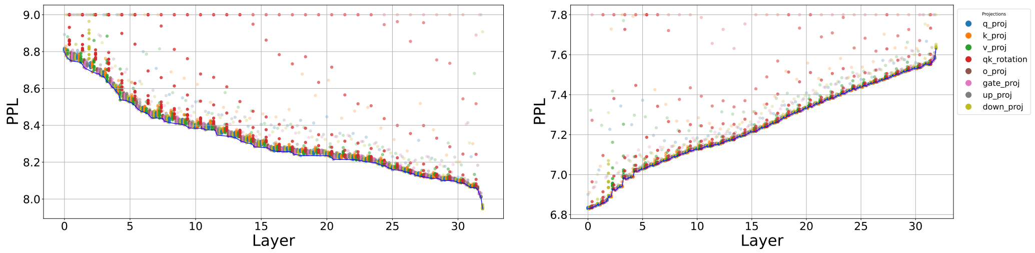
```
Require: A prime number $p$
Ensure: A Hadamard matrix $H$ of order $p+1$
$n \leftarrow p+1$                                        ▷ Determine the order of the matrix
Initialize $H$ as a $n \times n$ matrix with all entries set to 1   ▷ Start with a matrix of all ones
for $i \leftarrow 1$ to $n-1$ do                         ▷ Loop over rows (except the first row)
    $H[i,0] \leftarrow -1$                 ▷ Set the first column entry to -1 for current row
    $H[0,i] \leftarrow -1$                 ▷ Set the first row entry to -1 for current column
    for $j \leftarrow 1$ to $n-1$ do       ▷ Loop over columns (except the first column)
        if $i = j$ then
            $H[i,j] \leftarrow -1$                                ▷ Set diagonal entries to -1
        else
            $H[i,j] \leftarrow$ legendre_symbol$((i-1)-(j-1), p)$
        end if
    end for
end for
return $H$                                         ▷ Return the constructed Hadamard matrix
```

C Gradual Binary Search process

In this analysis, we examine the evolution of clipping ratios through GBS to better understand the dynamics of these parameters in LLMs. Figure 4 illustrates the perplexity (PPL) evolution during the optimization of a LLaMA3-8B model quantized to 4 bits. The graph displays the various tested values for each projection, optimized under two different configurations: starting the model in FP16 (blue line) and initiating the process in 4 bits. It is clear that starting in FP16 yields a better PPL on the training set of WikiText2, achieving 7.62, compared to starting in 4 bits, which results in a PPL of 7.94. On the test set we have the same dynamic with a PPL of 7.4 starting in FP16 and 7.69 starting in 4 bits.

Figure 5 illustrates the final configuration achieved by GBS for the same architecture, starting the process in both FP16 and 4-bit precision. It is clear that initiating in 4 bits results in a significantly more unstable configuration compared to starting in FP16. Many values remain at 1, and there is a high variance, indicating that the algorithm struggles to find a stable configuration when the entire model is in 4 bits. It also appears to have difficulty understanding the impact of small changes in the clipping ratio.

In contrast, starting in FP16 results in a stable configuration for every projection, with distinct dynamics. For instance, the qk_rotation projection exhibits minimal changes in the clipping ratio, with most layers close to 1. Conversely, the o_proj projection has values below 0.3, suggesting

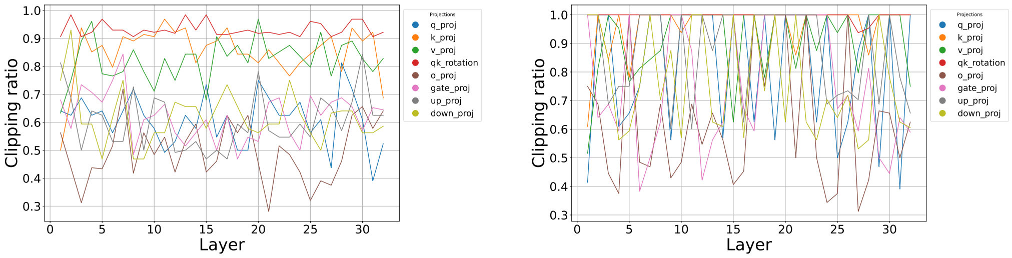


(a) GBS starting in 4 bits

(b) GBS starting in FP16

Figure 4: PPL vs Layer during Gradual Binary Search on 10% of Train WikiText2 for a LLaMA3-8B in 4-bit quantization and rotated with QuaRot. For better visualization we set a maximum PPL to 9. Points opacity represents the clipping ratio, the value is closer to 0 as transparency increases

that clipping to 30% of the maximum value can enhance performance. This figure underscores the importance of GBS in improving the model's quality by identifying the optimal clipping configuration, which is clearly not only ones.



(a) GBS starting in FP16

(b) GBS starting in 4 bits

Figure 5: Final configurations obtained with GBS started in 4 bits and in FP16 for a LLaMA3-8B

D Perplexity as objective

Perplexity is the central part of our optimization, it drives our search and it is supposed to reach a configuration which will performs better than all others with a bigger PPL. In figure 6 we can see how the average value on 6 benchmarks evolves with the perplexity. It clearly appears that a smaller PPL usually represents a better AVG.

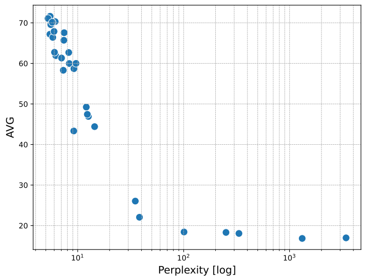


Figure 6: AVG over Perplexity for all results obtained in Tab 2 and 1

E More results

E.1 SpinQuant

Table 3: Results in 4 bits WAKV quantization on perplexity (PPL), PIQA, hellaswag (HS), arc-easy (ARC-E), arc-challenge (ARC-C), winogrande (WINO) and lambada, we also compute the average value (AVG) which represents a % of success. We compare our method, GBS, with SpinQuant and clearly observe that GBS outperforms SpinQuant across almost all computed metrics.

Model	Method	PPL↓	PIQA	HS	ARC-E	ARC-C	Wino	Lambada	AVG↑
Mistral 7B Inst v0.3	FP16	5.49	71.27	74.6	67.94	74.27	58.96	82.66	71.62
	SpinQuant	5.88	69	72.66	65.34	69.93	55.12	78.41	68.41
	SpinQuant + GBS	**5.83**	**69.01**	**72.66**	**65.36**	**72.14**	**56.91**	**79.5**	**69.26**
Mistral 7B v0.1	FP16	5.25	72.49	75.59	69.4	73.95	54.86	80.18	71.08
	SpinQuant	5.71	69.55	**73.45**	65.65	69.38	**48.98**	**76.98**	67.33
	SpinQuant + GBS	**5.62**	**70.02**	73.37	**66.66**	**71.19**	48.29	76.47	**67.67**
Llama2 7B	FP16	5.47	71.08	73.9	68.25	68.98	46.33	74.58	67.19
	SpinQuant	6.57	61.73	68.46	55.0	63.46	40.61	67.63	59.48
	SpinQuant + GBS	**6.15**	**63.24**	**69.01**	**57.46**	**65.04**	**41.04**	**68.43**	**60.7**
Llama3 8B	FP16	6.13	72.62	76.01	69.22	72.93	53.41	77.69	70.3
	SpinQuant	7.97	**65.11**	**69.01**	**61.21**	66.61	45.22	72.52	63.28
	SpinQuant + GBS	**7.69**	63.84	67.84	59.83	**70.4**	**47.35**	**73.36**	**63.77**

Table 4: Results in 3 bits WAKV quantization on perplexity (PPL), PIQA, hellaswag (HS), arc-easy (ARC-E), arc-challenge (ARC-C), winogrande (WINO) and lambada, we also compute the average value (AVG) which represents a % of success. We compare our method, GBS, with SpinQuant and clearly observe that GBS outperforms SpinQuant across almost all computed metrics.

Model	Method	PPL↓	PIQA	HS	ARC-E	ARC-C	Wino	Lambada	AVG↑
Mistral 7B Inst v0.3	FP16	5.49	71.27	74.6	67.94	74.27	58.96	82.66	71.62
	SpinQuant	14.31	20.39	28.39	12.38	49.72	24.57	40.57	29.34
	SpinQuant + GBS	**8.11**	**52.93**	**58.99**	**46.87**	**60.06**	**40.78**	**66.67**	**54.38**
Mistral 7B v0.1	FP16	5.25	72.49	75.59	69.4	73.95	54.86	80.18	71.08
	SpinQuant	18.88	16.58	21.97	11.2	51.3	24.23	37.67	27.16
	SpinQuant + GBS	**7.98**	**52.81**	**60.59**	**45.04**	**59.27**	**35.67**	**63.68**	**52.84**
Llama2 7B	FP16	5.47	71.08	73.9	68.25	68.98	46.33	74.58	67.19
	SpinQuant	425.2	0.24	0.45	0.04	**51.46**	28.33	27.57	18.02
	SpinQuant + GBS	**15.65**	**20.77**	**26.51**	**15.04**	50.67	**26.19**	**42.93**	**30.35**
Llama3 8B	FP16	6.13	72.62	76.01	69.22	72.93	53.41	77.69	70.3
	SpinQuant	316.6	2.71	3.1	2.31	50.51	22.35	29.0	18.33
	SpinQuant + GBS	**20.26**	**21.42**	**23.95**	**18.9**	**54.38**	**26.96**	**43.1**	**31.45**

E.2 DFRot

Table 5: Results in 4 bits WAKV quantization on perplexity (PPL), PIQA, hellaswag (HS), arc-easy (ARC-E), arc-challenge (ARC-C), winogrande (WINO) and lambada, we also compute the average value (AVG) which represents a % of success. We compare our method, GBS, with DFRot and clearly observe that GBS outperforms DFRot across all computed metrics.

Model	Method	PPL↓	PIQA	HS	ARC-E	ARC-C	Wino	Lambada	AVG↑
Mistral 7B Inst v0.3	FP16	5.49	71.27	74.6	67.94	74.27	58.96	82.66	71.62
	DFRot	5.94	68.11	79.43	68.5	72.42	64.58	**80.3**	72.22
	DFRot + GBS	**5.81**	**71.35**	**81.23**	**69.3**	**73.18**	**65.42**	80.13	**73.43**
Mistral 7B v0.1	FP16	5.25	72.49	75.59	69.4	73.95	54.86	80.18	71.08
	DFRot	5.75	**71.03**	78.94	69.83	74.03	65.63	78.28	72.96
	DFRot + GBS	**5.62**	70.88	**80.9**	**70.93**	**75.0**	**66.85**	**78.85**	**73.9**
Llama2 7B	FP16	5.47	71.08	73.9	68.25	68.98	46.33	74.58	67.19
	DFRot	6.23	65.04	76.66	65.75	69.47	62.02	72.61	68.59
	DFRot + GBS	**6.05**	**65.67**	**77.75**	**66.41**	**69.63**	**63.19**	**72.74**	**69.23**
Llama3 8B	FP16	6.13	72.62	76.01	69.22	72.93	53.41	77.69	70.3
	DFRot	7.95	68.11	76.01	64.92	68.5	61.34	74.17	68.84
	DFRot + GBS	**7.56**	**72.53**	**76.82**	**66.35**	**69.53**	**63.17**	**75.0**	**70.57**

Table 6: Results in 3 bits WAKV quantization on perplexity (PPL), PIQA, hellaswag (HS), arc-easy (ARC-E), arc-challenge (ARC-C), winogrande (WINO) and lambada, we also compute the average value (AVG) which represents a % of success. We compare our method, GBS, with DFRot and clearly observe that GBS outperforms DFRot across all computed metrics.

Model	Method	PPL↓	PIQA	HS	ARC-E	ARC-C	Wino	Lambada	AVG↑
Mistral 7B Inst v0.3	FP16	5.49	71.27	74.6	67.94	74.27	58.96	82.66	71.62
	DFRot	11.26	53.28	67.57	35.6	40.33	30.88	57.57	47.54
	DFRot + GBS	**7.58**	**63.22**	**75.35**	**59.32**	**65.19**	**53.46**	**71.73**	**64.71**
Mistral 7B v0.1	FP16	5.25	72.49	75.59	69.4	73.95	54.86	80.18	71.08
	DFRot	13.63	55.01	65.45	27.89	32.52	23.25	50.63	42.46
	DFRot + GBS	**7.64**	**62.83**	**73.72**	**56.99**	**64.33**	**49.64**	**68.12**	**62.6**
Llama2 7B	FP16	5.47	71.08	73.9	68.25	68.98	46.33	74.58	67.19
	DFRot	26.64	49.96	58.81	13.19	14.81	11.57	39.14	31.25
	DFRot + GBS	**10.96**	**56.12**	**66.81**	**34.43**	**40.4**	**28.45**	**55.01**	**46.87**
Llama3 8B	FP16	6.13	72.62	76.01	69.22	72.93	53.41	77.69	70.3
	DFRot	140.78	52.41	54.62	2.82	3.38	2.27	31.85	24.56
	DFRot + GBS	**22.14**	**54.85**	**61.92**	**25.53**	**29.58**	**21.48**	**48.67**	**40.34**

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 wrote our abstract at the end so every claim in the abstract perfectly represents what we showed in the paper

   Guidelines:

   - The answer NA 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 NA 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 have a section dedicated to limitations in this paper

   Guidelines:

   - The answer NA 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: In this paper we proved the efficiency of Hadamard matrices to reduce outliers in a tensor over random orthogonal matrices and we linked it to dimension of tokens. The proof is clearly explained in appendix A by using 2 established theorems of high dimension theory and extreme values theory which can also be added if needed.

Guidelines:

- The answer NA 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: Every details to reproduce our results are clearly explained in the paper, especially is section 5.1 where we explain every parameter to configure. Every dataset, models and benchmarks can be accessed easily and freely. Our main contribution GBS is clearly explained and can be implemented by the reader if needed.

Guidelines:

- The answer NA 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[No]

   Justification: We can't publish our code yet due to institutional constraints but we plan to release it if the paper is accepted.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Every hyper parameter setting is explained in section 5.1 and the base code QuaRot is accessible online.

   Guidelines:

   - The answer NA 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: We don't report error bars due to computational cost but we provide a significant number of experiments to support our claims.

   Guidelines:

   - The answer NA 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: We provide this information in section 5.1 where we explain we only use one GPU A100 for all our experiments.

   Guidelines:

   - The answer NA 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: Our paper respect every aspect of these ethics guidlines

   Guidelines:

   - The answer NA 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: [NA]

    Justification: This paper doesn't have any societal impact

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: In our paper we use already available open models and datasets and we don't modify their architecture.

Guidelines:

- The answer NA 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: Every dataset or models are correctly credited in this paper and available on huggingface.

Guidelines:

- The answer NA 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: Every assets we used can be accessed freely online.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: We don't involve human subjects in this paper.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: We don't involve human subjects in this paper.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We used LLMs only for formatting purpose

Guidelines:

- The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.