HOFT: Householder Orthogonal Fine-tuning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Adaptation of foundation models using low-rank methods is a widespread approach. Another way to adapt these models is to employ orthogonal fine-tuning methods, which are less time and memory efficient despite their good generalization properties. In this work, we propose Householder Orthogonal Fine-tuning (HOFT), a novel orthogonal fine-tuning method that aims to alleviate the time and space complexity. Moreover, some theoretical properties of the orthogonal fine-tuning paradigm are explored. From this exploration, Scaled Householder Orthogonal Fine-tuning (SHOFT) is proposed. Both HOFT and SHOFT are evaluated in downstream tasks, namely commonsense reasoning, machine translation, subject-driven generation and mathematical reasoning. Compared with state-of-the-art adaptation methods, HOFT and SHOFT show comparable or better results.

1 Introduction

Nowadays, fine-tuning foundation models for downstream tasks [17] is the standard approach to model adaptation these days thanks to their knowledge across many domains. By tuning far fewer parameters than full fine-tuning using parameter-efficient fine-tuning techniques (PEFT), the model is able to learn the key aspects of a task and perform comparably or even better than full fine-tuning [18]. This fact makes PEFT methods a particularly well-suited approach for efficiently adapting these models. The employment of PEFT methods has enabled the adaptation of large foundation models without the necessity of compute-intensive hardware infrastructure, making adaptation accessible to a broader user community.

The most popular PEFT methods are based on low-rank approximations, including Low-Rank Adaptation (LoRA) [19] and Weight-Decomposed Low-Rank Adaptation (DoRA) [27]. These methods work under the assumption that the learnable parameters must reside in a lower intrinsic dimension [2]. Alternatively, there are methods proposing the use of orthogonal matrices for adaptation, such as Orthogonal Fine-tuning (OFT) [36] and Orthogonal Butterfly (BOFT) [29]. These methods hypothesize that a good fine-tuned model should have a minimal difference in hyperspherical energy compared to the pre-trained model [26, 28]. In brief, the assumption made is that orthogonality is required to learn new features while keeping pre-trained information [36, 53]. Whilst the performance of these techniques has been demonstrated, their runtime and memory footprint make them a less preferable option for use in real-world applications. A recent approach to balance low-rank and orthogonal methods is Householder Reflection Adaption (HRA) [53], which constrains orthogonality through the incorporation of a term within the loss function. With the employment of an additional weight λ for the orthogonality regularizer, HRA aims to construct the chained matrix product of Householder transformations [14].

Orthogonal fine-tuning methods generally result in the construction of a single orthogonal matrix for adaptation purposes. This work demonstrates that two orthogonal matrices are required in order to ensure full expressivity in orthogonal fine-tuning methods. This leads us to propose **Householder Orthogonal Fine-tuning** (HOFT): a novel orthogonal fine-tuning technique using two orthogonal

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

matrices as directional components efficiently updated through orthogonal transformations. For efficiency, these matrices are obtained by accumulating Householder transformations via the CWY transform [25, 21] along with a fast inverse approximation. Additionally, we draw inspiration from DoRA's analysis, which shows that fine-tuning magnitude and direction separately closely matches the learning dynamics of full fine-tuning. From this, a variant of HOFT incorporating an additional scaling transformation is proposed: **Scaled Householder Orthogonal Fine-tuning** (SHOFT).

In order to evaluate both methods, a series of experiments are conducted in four distinct areas: commonsense reasoning, machine translation, subject-driven generation and mathematical reasoning. The selection of these tasks was made with the intention of evaluating the efficacy of the proposed methods along with both low-rank and orthogonal PEFT. Notably, quantized models are also adapted in mathematical reasoning experiments. Experimental results demonstrate that HOFT and SHOFT benefit from retaining the relational structure of pre-trained weights, reaching or exceeding the performance of existing state-of-the-art PEFT baselines.

2 Related Work

Low-Rank Adaptation Methods in this family assume that effective fine-tuning updates lie on a compact, low-dimensional manifold [19, 27, 23, 55, 24, 51, 20, 46]. LoRA [19] introduces trainable low-rank adapter matrices into each Transformer layer, freezing the original weights and reducing trainable parameters by several orders of magnitude. DoRA [27] separates the fine-tuning of directional and scaling components by normalizing LoRA's output and applying a scaling transformation. PiSSA [30] employs singular value decomposition (SVD) on pre-trained weight matrices to initialize LoRA adapters in principal subspaces, maintaining most of the original model's expressive capacity. QLoRA [12] combines 4-bit NormalFloat (NF4) quantization with LoRA, enabling the fine-tuning of 65B-parameter models on a single 48GB GPU while preserving near full-precision quality. QA-LoRA [49] employs group-wise quantization operators to selectively compress adapter updates with minimal impact on task performance loss.

Orthogonal Fine-Tuning Orthogonal fine-tuning methods learn distance preserving transformations in weight space, keeping geometric properties such as hyperspherical energy among neuron activations [36, 29]. Previous works show how the imposition of orthogonality constraints within deep learning architectures is conducive to enhancing performance [5, 44, 48, 13, 3]. OFT [36] employs Cayley parameterization [22] to generate orthogonal matrix blocks. Additionally, COFT [36] constrains the orthogonal matrix to be within a small neighborhood of the pre-trained matrix. BOFT [29] reduces OFT parameter footprint by factorizing orthogonal updates into butterfly structures inspired by the Cooley–Tukey FFT algorithm [6], achieving similar generalization gains with fewer trainable parameters. The employment of hybrid methods, such as HRA, enforces hyperspherical constraints on low-rank adapters to blend both paradigms via a term in the loss function controlled by a weight [53].

3 Proposed Method

3.1 Orthogonal fine-tuning paradigm

As discussed in Section 2, orthogonal fine-tuning stresses the importance of preserving the hyperspherical energy of the given matrix $\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top \in \mathbb{R}^{m\times n}$. Although it is clear that this can be done by adapting both singular vector matrices $\mathbf{U}$ and $\mathbf{V}$, it is common practice to keep $\mathbf{V}^\top$ unchanged and adapt only $\mathbf{U}$ [36, 29, 53].

Consider all possible orthogonal transformations of M into an adapted matrix $\widehat{\mathbf{M}} = \widehat{\mathbf{U}}\widehat{\boldsymbol{\Sigma}}\widehat{\mathbf{V}}^\top$ preserving its hyperspherical energy; that is, meaning that $\widehat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}$, though $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{V}}^\top$ might differ from $\mathbf{U}$ and $\mathbf{V}^\top$ respectively. Suppose there exists an orthogonal matrix $\mathbf{Q} \in \mathrm{O}(m)$ such that $\widehat{\mathbf{M}} = \mathbf{Q}\mathbf{M}$, that is $\widehat{\mathbf{U}}\widehat{\boldsymbol{\Sigma}}\widehat{\mathbf{V}}^\top = \mathbf{Q}\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$. Since $\mathbf{Q}$ is arbitrary, we can set $\mathbf{Q} = \widehat{\mathbf{U}}\mathbf{U}^\top$, and due to hyperspherical energy conservation, $\widehat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}$. However, we cannot ensure that $\mathbf{V}$ and $\widehat{\mathbf{V}}$ are equal. Thus, in order to cover all possible adapted matrices, we need two orthogonal matrices $\mathbf{Q_U} \in \mathrm{O}(m), \mathbf{Q_V} \in \mathrm{O}(n)$. Only in this case we can ensure that it is possible to obtain $\widehat{\mathbf{M}}$, since we can set $\mathbf{Q_U} = \widehat{\mathbf{U}}\mathbf{U}^\top$ and $\mathbf{Q_V} = \mathbf{V}\widehat{\mathbf{V}}^\top$ to construct $\mathbf{Q_U}\mathbf{M}\mathbf{Q_V} = \widehat{\mathbf{M}}$.

In terms of approximation error, pre- and post-multiplying the pre-trained matrix $\mathbf{M}$ by distance-preserving transformations exactly captures all adapted matrices $\widehat{\mathbf{M}}$ that maintain the same hyperspherical energy. However, the error incurred when applying just one orthogonal matrix leads us to a known problem, the Orthogonal Procrustes Problem [15], which has a solution if $\widehat{\mathbf{M}}$ and $\mathbf{M}$ are known matrices. In this one-transform setting, a theoretical upper bound on the Frobenius norm error is given by

$$\min_{\mathbf{Q}\in \mathrm{O}(m)} \left\| \widehat{\mathbf{M}} - \mathbf{Q}\mathbf{M} \right\|_F \leq 2\sqrt{m}\, \|\mathbf{M}\|_F \tag{1}$$

Further details and the proof of Equation 1 are provided in Appendix C.

3.2 CWY transform and inverse approximation

As observed in [36, 29], computing parameterized orthogonal matrices is computationally costly though it can be sped up with numerical methods. In our case, the composition of multiple Householder transformations can be cast into high-performance matrix-matrix products through the WY and CWY transforms [21, 25]. The following result from [21] allows us to construct an orthogonal matrix by accumulating householder transformations:

Theorem 1 *Let the matrix $\mathbf{U} \in \mathbb{R}^{m\times r}$ have linearly independent columns. Partition $\mathbf{U}$ by columns as $\mathbf{U} = (\mathbf{u}_1 \mid \mathbf{u}_2 \mid \ \ldots \ \mid \mathbf{u}_r)$ and consider the vector $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_r)^\top$ with $\tau_i \neq 0, 1 \leq i \leq r$. Then, there exists a unique nonsingular upper triangular matrix $S \in \mathbb{R}^{r\times r}$ such that*

$$\mathbf{Q}_\mathbf{U} = \left(\mathbf{I} - \frac{\mathbf{u}_1\mathbf{u}_1^\top}{\tau_1}\right)\left(\mathbf{I} - \frac{\mathbf{u}_2\mathbf{u}_2^\top}{\tau_2}\right)\cdots\left(\mathbf{I} - \frac{\mathbf{u}_r\mathbf{u}_r^\top}{\tau_r}\right) = \mathbf{I} - \mathbf{U}\mathbf{S}^{-1}\mathbf{U}^\top \tag{2}$$

where $\mathbf{Q}_\mathbf{U} \in \mathrm{O}(m)$. $\mathbf{S}$ can be computed following two steps:

1. $\mathbf{S} :=$ *the upper triangular part of* $\mathbf{U}^\top\mathbf{U}$.
2. *Divide the diagonal elements of* $\mathbf{S}$ *by two.*

As in the case of many orthogonal parameterization methods [14], there is a matrix inverse to be computed. This fact makes orthogonal parameterization methods non-scalable, since the inverse computation during training and the gradient update computation are resource-intensive. However, in the case of the CWY transform, the inverse can be approximated with a high degree of precision. In order to efficiently compute $\mathbf{S}^{-1}$, Neuman Series are required [14]. We can separate $\mathbf{S} = \mathbf{D} + \mathbf{A} = \mathbf{D}(\mathbf{I} + \mathbf{D}^{-1}\mathbf{A})$ where $\mathbf{D}$ is a diagonal matrix and $\mathbf{A}$ is a strictly upper triangular matrix. The inverse will be:

$$\mathbf{S}^{-1} = \left(\mathbf{I} + \mathbf{D}^{-1}\mathbf{A}\right)^{-1}\mathbf{D}^{-1} = \left(\sum_{i=0}^{\infty}\left(-\mathbf{D}^{-1}\mathbf{A}\right)^i\right)\mathbf{D}^{-1} \approx \mathbf{D}^{-1} - \mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1} \tag{3}$$

It can be demonstrated that, since $\mathbf{A} \in \mathbb{R}^{r\times r}$ is strictly upper triangular, then the spectral radius $\rho\left(\mathbf{D}^{-1}\mathbf{A}\right)$ is less than one and we can ensure that the series from Equation 3 always converges. In fact, $\sum_{i=0}^{\infty}(-\mathbf{D}^{-1}\mathbf{A})^i = \sum_{i=0}^{r-1}(-\mathbf{D}^{-1}\mathbf{A})^i$, and the inverse approximation error grows with the number of columns r.

Taking the first and second term of the series in order to approximate the inverse only require diagonal inverses, which are very fast to compute. Rearranging Equation 2, the final equation to approximately compute the accumulated householder product is:

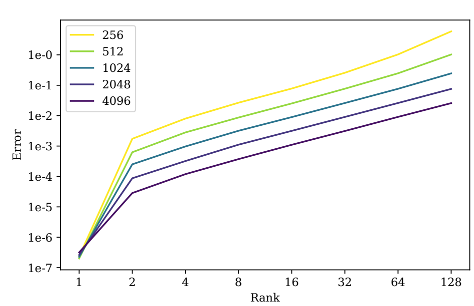


Figure 1: Inverse approximation error

$$\mathbf{Q_U} = \mathbf{I} - \mathbf{U}\mathbf{S}^{-1}\mathbf{U}^\top \approx \mathbf{I} + \mathbf{U}\left(\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1} - \mathbf{D}^{-1}\right)\mathbf{U}^\top \tag{4}$$

To empirically assess the error magnitude, we conducted an experiment approximating a random gaussian accumulated Householder transformation. Figure 1 illustrates how the inverse approximation error varies depending on the rank r. The error is defined as $\left\|\mathbf{I} - \mathbf{Q_U}\mathbf{Q_U}^\top\right\|_F / \sqrt{n}$, where n denotes the matrix dimension. As expected, the error is zero when $r = 1$, since the approximation is exact in that case. Although the error grows when increasing r, the growth rate remains modest. In particular, for $r \ll m$, the approximation remains remarkably accurate. Further details can be found in Appendix B.

3.3 Householder Orthogonal Fine-tuning

Given householder vectors stored in the columns of $\mathbf{U} \in \mathbb{R}^{m\times r}$ and $\mathbf{V} \in \mathbb{R}^{n\times r}$, we construct orthogonal matrices $\mathbf{Q_U} \in \mathrm{O}(m)$ and $\mathbf{Q_V} \in \mathrm{O}(n)$ by applying the CWY transform along with the inverse approximation of $\mathbf{S}$ from Section 3.2. As discussed in Section 3.1, the resulting matrix $\widehat{\mathbf{M}} = \mathbf{Q_U}\mathbf{M}\mathbf{Q_V}$ can express any matrix $\widehat{\mathbf{M}} \in \mathbb{R}^{m\times n}$ such that the hyperspherical energy remains the same as $\mathbf{M} \in \mathbb{R}^{m\times n}$. We call this novel method Householder Orthogonal Fine-tuning (HOFT). As illustrated in Figure 2, our method adapts both $\mathbf{U}, \mathbf{V}^\top$ while preserving the same hyperspherical energy.

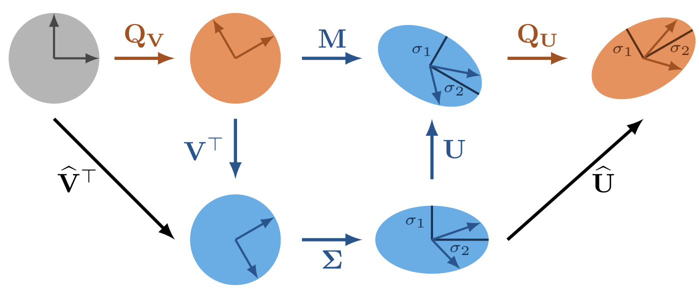


Figure 2: Diagram of our proposed HOFT method

Similar to HRA's rank r [53], HOFT also employs r householder vectors. For both inverse approximations, the computational complexity is $\mathcal{O}\left(r^2(m+n)\right)$, and the matrix-vector multiplications require $\mathcal{O}\left(2mr + 2nr + mn + 2r^2\right)$. Altogether, the total time complexity of HOFT is $\mathcal{O}\left(mn + 2r(m+n) + r^2(m+n+2)\right) \sim \mathcal{O}\left(mn + (m+n)(r^2+2r)\right)$. A comparison of the computational complexity of HOFT to other parameterized orthogonal-based methods is provided in Table 1.

Table 1: Comparisons of parameterized orthogonal-based methods

Method	#Parameters	Complexity	Coverage
OFT	$\frac{m(b-1)}{2}$	$\mathcal{O}(mn + m(b^2+b))$	$b = m$
BOFT	$\frac{mk(b-1)}{2}$	$\mathcal{O}(mn + mk(b^2+m))$	$k = \log m$ and $b = 2$
HRA	rm	$\mathcal{O}(mn + mr)$	$r = m - 1$
HOFT	$r(m+n)$	$\mathcal{O}(mn + (m+n)(r^2+2r))$	$r = \max(m,n) - 1$

One drawback of OFT is that it requires b to be large in order to achieve $\mathrm{O}(m)$ coverage [36]. The increase of b cannot be arbitrary because of the cost of inverting $b\times b$ matrices. BOFT, on the other hand, offers better coverage at the expense of higher time complexity [29]. HRA provides even better coverage than the two previous methods; however, its Householder transformations must be applied sequentially, and when $\lambda = \infty$, its runtime matches that of OFT [53]. By contrast, HOFT provides the same coverage as HRA, and because most of its computations can be parallelized, it achieves greater speedup and represents an attractive alternative.

Although LoRA and DoRA can be randomly initialized, OFT and BOFT cannot due to the necessity of preserving orthogonality; Cayley's parameterization [14] needs skew-symmetric matrix $\mathbf{R} = \mathbf{0}$ to ensure that the orthogonal parameterized matrix is $\mathbf{Q} = \mathbf{I}$. In general, orthogonal PEFT methods cannot be randomly initialized. However, HOFT and SHOFT can be randomly initialized by considering consecutive pairs of equal vectors $\mathbf{u}_i$. Since they express the same reflection, we can place Householder vectors in the form $\mathbf{U} = (\mathbf{u}_1 \mid \mathbf{u}_1 \mid \cdots \mid \mathbf{u}_k \mid \mathbf{u}_k)$, which yields the identity matrix:

$$\mathbf{Q}_\mathbf{U} = \underbrace{\left(\mathbf{I} - \frac{\mathbf{u}_1\mathbf{u}_1^\top}{\tau_1}\right)\left(\mathbf{I} - \frac{\mathbf{u}_1\mathbf{u}_1^\top}{\tau_1}\right)}_{\mathbf{I}} \cdots \underbrace{\left(\mathbf{I} - \frac{\mathbf{u}_k\mathbf{u}_k^\top}{\tau_k}\right)\left(\mathbf{I} - \frac{\mathbf{u}_k\mathbf{u}_k^\top}{\tau_k}\right)}_{\mathbf{I}} = \mathbf{I} \tag{5}$$

Thus, if r is even, we can generate $k = \frac{r}{2}$ pairs of random vectors. If r is odd, we can generate $k = \lfloor \frac{r}{2} \rfloor$ pairs of random vectors and a zero vector. Vectors $\mathbf{u}_i$ are picked from a high-dimensional gaussian distribution. $\mathbf{V}$ is also initialized following this procedure, making $\mathbf{Q}_\mathbf{V} = \mathbf{I}$ at the beginning of the training.

3.4 Scaled Householder Orthogonal Fine-tuning

The use of scaling transformations in orthogonal fine-tuning methods has been studied in [36] as a way to improve their performance. Drawing also inspiration from DoRA's weight decomposition analysis [27], we propose a variant of HOFT that employs a scaling transformation: Scaled Householder Orthogonal Fine-tuning (SHOFT). As observed in Section 3.1, placing the scaling transformation near the singular value matrix will be interesting from a SVD perspective. Since scaling is performed between two distance preserving transformations, the effect of $\mathbf{m}$ in the singular values of $\mathbf{M}$ is closely controlled. Thus, SHOFT formulation will be as follows

$$\widehat{\mathbf{M}} = \mathbf{Q}_\mathbf{U}\mathbf{m}\mathbf{M}\mathbf{Q}_\mathbf{V} = \mathbf{Q}_\mathbf{U}\mathbf{m}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{Q}_\mathbf{V} \tag{6}$$

where $\mathbf{Q}_\mathbf{U}, \mathbf{Q}_\mathbf{V}$ and $\mathbf{m}$ are formed by trainable parameters. It seems more intuitive to be able to redirect with $\mathbf{Q}_\mathbf{V}$, transform with $\mathbf{M}$, then scale with $\mathbf{m}$ and finally redirect with $\mathbf{Q}_\mathbf{U}$. SHOFT is more flexible since it is no longer constrained to keep the same hyperspherical energy. All elements of vector $\mathbf{m}$ are initialized to one. As observed in other PEFT methods [27, 36], the increase on the amount of trainable parameters due to adding a magnitude vector $\mathbf{m} \in \mathbb{R}^m$ is marginal.

4 Experiments

In order to compare HOFT and SHOFT along with other PEFT methods, four main tasks have been selected: commonsense reasoning, machine translation, subject-driven generation and mathematical reasoning. In these tasks, state-of-the-art PEFT methods are evaluated using different pre-trained models to show robustness along different architectures. In addition, quantized models are also employed for evaluating mathematical reasoning. All hyperparameter settings used in the experiments are provided in Appendix A. Additionally, an empirical comparison of time and memory complexity is given in Appendix D.

4.1 Commonsense reasoning

For measuring commonsense reasoning performance, we compare HOFT and SHOFT with DoRA and LoRA across eight standard commonsense reasoning benchmarks: BoolQ [8], PIQA [4], SIQA [43], HellaSwag [54], WinoGrande [42], ARC-e [9], ARC-c [9] and OBQA [31]. Following DoRA [27], the training splits of all eight tasks are merged into a single training set, and then each model is evaluated separately on the original test set of each task. The models employed are LLaMA3.1-8B [16], Qwen2.5-7B [50], Phi4-14B [1], and Qwen2.5-14B [50]. We initialize DoRA [27] and LoRA [19] using PiSSA [30]. We set $r = 16$ for all PEFT methods and train the models for two epochs.

The results of each individual task along with the average task accuracy per model and PEFT method are shown in Table 2, where it can be seen that HOFT and SHOFT generally achieve higher scores than LoRA and DoRA across most models, with SHOFT performing comparably to DoRA for

Qwen2.5-7B. Moreover, both HOFT and SHOFT continue to deliver strong results as model size grows, demonstrating solid performance on both Phi4-14B and Qwen2.5-14B. In particular, HOFT and SHOFT attain the highest scores on nearly every task, matching LoRA and DoRA only on PIQA and ARC-e. This underscores their robustness and efficiency when trained on datasets containing multiple domains.

Table 2: Accuracy comparison (%) on various commonsense reasoning benchmarks

Model	Method	#Params (%)	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg.
LLaMA3.1-8B	LoRA	0.35	88.2	**88.5**	80.3	96.7	80.5	91.9	82.3	87.4	87.0
	DoRA	0.36	88.1	89.1	80.1	96.6	**81.4**	92.0	82.5	86.8	87.1
	HOFT	0.35	88.5	**88.5**	**80.9**	**96.8**	80.4	**92.7**	**83.2**	**88.4**	**87.4**
	SHOFT	0.36	**88.8**	**88.5**	80.1	**96.8**	81.2	92.0	82.9	86.6	87.1
Qwen2.5-7B	LoRA	0.35	88.4	89.5	**79.6**	**96.8**	**82.5**	95.8	88.7	92.2	89.2
	DoRA	0.36	88.9	**89.8**	79.2	**96.8**	**82.5**	**96.2**	88.9	92.4	**89.3**
	HOFT	0.35	**89.0**	89.1	79.2	96.4	80.4	95.9	88.4	92.4	88.9
	SHOFT	0.36	88.8	89.5	79.5	96.5	80.7	95.7	**89.1**	**93.4**	89.2
Phi4-14B	LoRA	0.33	89.7	92.0	81.7	97.3	**87.9**	97.9	93.1	94.2	91.7
	DoRA	0.35	90.0	91.9	82.0	**97.4**	87.3	98.0	93.5	94.0	91.8
	HOFT	0.33	**90.1**	**92.7**	**82.3**	**97.4**	86.7	**98.1**	94.3	93.6	91.9
	SHOFT	0.35	90.0	**92.7**	81.9	97.3	87.4	98.0	**94.5**	**95.4**	**92.2**
Qwen2.5-14B	LoRA	0.31	89.9	**92.7**	82.1	98.0	87.1	**98.1**	93.6	95.0	92.1
	DoRA	0.32	89.9	92.5	82.6	98.0	87.3	**98.1**	93.0	94.6	92.0
	HOFT	0.31	90.2	91.9	**83.8**	98.0	87.6	97.7	**93.7**	**96.2**	**92.4**
	SHOFT	0.32	**90.3**	92.3	83.0	**98.1**	**88.2**	97.2	92.7	**96.2**	92.3

4.2 Machine Translation

For measuring machine translation performance, HOFT and SHOFT are compared with DoRA and LoRA using four languages from the CoVoST 2 [47] dataset: Slovene, German, Latvian and French. We chose these languages in order to have two well-represented languages and two low-resource languages. For French and German, models are trained on the first 10K elements of the training split. Three models are adapted for this task: NLLB-3.3B [11], LLaMA2-7B [45], and LLaMA3.1-8B [16]. We set $r = 16$ for all PEFT methods and train the models for 2 epochs. Both BLEU [33, 35] and COMET [39, 38] results are provided for each individual language per model and PEFT method. Results obtained are shown in Table 3. We additionally provide baseline performance of the models.

Table 3: Performance comparison on X $\rightarrow$ English machine translation tasks

Model	Method	#Params (%)	Slovene		German		Latvian		French	
			BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET
NLLB-3.3B	Baseline	-	39.7	87.5	39.3	86.2	31.2	81.3	38.5	84.9
	LoRA	0.42	46.8	89.2	44.5	**87.7**	38.2	83.9	**49.7**	**87.8**
	DoRA	0.43	46.8	89.1	**44.7**	87.6	38.2	83.9	49.5	87.7
	HOFT	0.42	**48.0**	89.4	44.4	87.6	38.6	83.9	49.5	87.7
	SHOFT	0.43	46.4	**89.5**	44.5	**87.7**	**38.7**	**84.0**	**49.7**	**87.8**
LLaMA2-7B	0-shot	-	26.8	72.8	30.4	74.1	4.5	52.2	37.2	79.3
	LoRA	0.19	39.3	84.7	41.5	86.9	15.5	66.2	47.0	87.2
	DoRA	0.19	39.6	84.8	41.4	86.9	**16.2**	**66.6**	47.0	87.2
	HOFT	0.19	40.6	85.2	41.4	86.9	15.8	**66.6**	47.0	**87.3**
	SHOFT	0.19	**41.2**	**85.6**	**41.6**	**87.0**	15.7	65.9	**47.1**	**87.3**
LLaMA3.1-8B	0-shot	-	34.2	77.8	40.9	86.2	22.9	70.8	41.6	82.7
	LoRA	0.12	36.2	84.1	42.3	87.4	32.7	**80.9**	**46.8**	85.5
	DoRA	0.12	42.4	85.0	42.2	87.4	**32.8**	80.8	46.7	85.5
	HOFT	0.12	**44.2**	**86.6**	42.9	87.5	32.2	80.4	46.7	**85.6**
	SHOFT	0.12	43.6	86.4	**43.1**	**87.7**	31.9	80.4	**46.8**	**85.6**

From Table 3 we can observe how HOFT and SHOFT provide competitive results in French and German. In Latvian, HOFT and SHOFT give similar results in the case of NLLB-3.3B. For Slovene, both methods clearly outperform LoRA and DoRA with LLaMA2-7B, while HOFT in BLEU and SHOFT in COMET with NLLB-3.3B. Notably, the difference on both metrics is significantly higher with LLaMA3.1-8B. Overall, the top BLEU and COMET scores are almost always achieved by either HOFT or SHOFT, underlining their effectiveness across multiple languages.

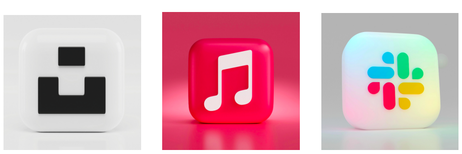
4.3 Subject-driven generation

For subject-driven generation, we follow the experimental protocol of HRA [53], using the DreamBooth dataset [41] to train and evaluate on 25 distinct subjects, each with 30 associated prompts. We adapt the pre-trained Stable Diffusion (SD) model [40] and compare PEFT methods quantitatively across four metrics: subject fidelity (DINO [7] and CLIP-I [37]), prompt fidelity (CLIP-T [37]), and sample diversity (LPIPS [56]).

Table 4: Quantitative comparison of subject-driven generation

Method	#Param (M)	DINO ↑	CLIP-I ↑	CLIP-T ↑	LPIPS ↑
Real Images	–	0.764	0.890	–	0.562
DreamBooth	859.52	0.614	0.778	0.239	0.737
LoRA	0.80	0.613	0.765	0.237	0.744
COFT$_{b=4}$	23.3	0.630	0.783	0.235	0.744
OFT$_{b=4}$	23.3	0.632	0.785	0.237	0.746
HRA$_{r=7,8,\lambda=0}$	0.69	0.670	0.803	0.238	0.758
HRA$_{r=7,8,\lambda=10^{-3}}$	0.69	0.661	0.799	0.255	0.760
HRA$_{r=7,8,\lambda=\infty}$	0.69	0.651	0.794	**0.274**	**0.778**
HOFT$_{r=2}$	0.40	0.657	0.793	0.239	0.758
SHOFT$_{r=2}$	0.41	0.658	0.793	0.241	0.757
HOFT$_{r=4}$	0.80	**0.680**	**0.810**	0.235	0.752
SHOFT$_{r=4}$	0.81	**0.680**	0.808	0.235	0.747

The results, together with the provided baselines, are summarized in Table 4. Both HOFT and SHOFT outperform all baselines in subject fidelity. In terms of textual prompt fidelity, they achieve results comparable with LoRA, OFT, and COFT. For sample diversity, they also deliver competitive performance. Additionally, we also tested HOFT and SHOFT at half the rank. Even with fewer trainable parameters, both methods consistently outperform LoRA, OFT, and COFT across all metrics, while remaining competitive with HRA on subject fidelity.



images of 3d icons

images of lego sets

Figure 3: Examples of training images of 3D icons and lego sets

Therefore, in order to gain a deeper insight into subject fidelity, we conducted an additional experiment following DoRA [27]. We fine-tuned a pre-trained Stable Diffusion XL (SDXL) model [34] on two datasets: 3D icons and lego sets. In Figure 3 we can see some examples of the styles to be learned. In this experiment, five PEFT methods are used for evaluation: LoRA, HRA, OFT, HOFT, and SHOFT. To ensure a fair comparison, all methods used the same random sample seed for generating the images.

As shown in Figure 4, HOFT and SHOFT provide better personalization than LoRA, HRA, and OFT. When generating 3D icons, both methods closely match the style and subject of the training images. This highlights the value of orthogonality: while OFT also produces competitive results, LoRA and HRA struggle to generate realistic 3D icons. Moreover, HOFT and SHOFT produce accurate text in the lego sets, while the rest do not achieve it. Additional qualitative examples can be found in Appendix E.

Prompt: a TOK 3d icon of an orange llama eating ramen, in the style of TOK

Prompt: a TOK lego set of an orange llama eating ramen, in the style of TOK

LoRA HRA OFT HOFT SHOFT

Figure 4: Qualitative results on lego sets and 3d icons datasets

4.4 Mathematical reasoning

For the mathematical reasoning experiments, we follow the HRA guidelines [53]. We fine-tune LLaMA2-7B [45] on the MetaMathQA dataset [52], which contains a diverse amount of mathematical questions along with rationalized answers. HOFT and SHOFT are evaluated on the GSM8K [10] and MATH [52] validation sets. Table 5 shows the accuracy of these methods alongside other PEFT baselines.

Table 5: Accuracy comparison (%) on mathematical reasoning datasets

Method	GSM8K	MATH
Baseline	14.6	2.5
LoRA	50.2	7.8
OFT	50.1	8.4
BOFT	50.6	8.6
PiSSA	53.1	7.4
HRA	56.3	9.3
HOFT	**56.6**	8.9
SHOFT	55.0	**9.8**

The results in Table 5 show that HOFT and SHOFT are competitive with existing PEFT methods on mathematical reasoning benchmarks. HOFT achieves the highest accuracy on GSM8K, while SHOFT achieves the best score on the more challenging MATH dataset. This suggests that the scaling transformation plays a role to improve performance on harder math questions.

4.5 QHOFT: Quantized HOFT

In addition to the previous mathematical reasoning experiment, two additional experiments are performed in order to test the quantized versions of HOFT and SHOFT. We adapt 4-bit quantized [12] LLaMA2-7B [45] and LLaMA3.1-8B [16] to GSM8K [10] and Orca-Math [32] datasets separately and evaluate them on their respective test datasets. In particular, we follow DoRA [27] Orca-Math experimental setup: 100K elements for training and 2K for evaluation. The experimental results are reported in Table 6.

Table 6: Accuracy comparison (%) on mathematical reasoning datasets using quantized models

Model	Method	#Params (%)	GSM8K	Orca-Math
LLaMA2-7B	QLoRA	0.19	27.9	14.4
	QDoRA	0.19	29.0	13.0
	QHOFT	0.19	**30.5**	14.7
	QSHOFT	0.19	29.3	**15.5**
LLaMA3.1-8B	QLoRA	0.12	53.8	54.1
	QDoRA	0.12	56.5	53.8
	QHOFT	0.12	55.0	**57.2**
	QSHOFT	0.12	**57.0**	54.6

The results in Table 6 demonstrate that the quantized versions of HOFT and SHOFT consistently outperform QLoRA and QDoRA under extreme parameter efficiency. On LLaMA2-7B, QHOFT achieves the highest GSM8K accuracy, while QSHOFT leads on Orca-Math. On the larger LLaMA3.1-8B model, QSHOFT delivers the best GSM8K performance, and QHOFT achieves the best Orca-Math score. These results confirm that QHOFT and QSHOFT perform well even with aggressive 4-bit quantization.

5 Limitations

One limitation of our work is the challenge of adapting architectures with low-dimensional weight matrices: neither HOFT nor SHOFT can fully enforce orthogonality in their learned weights when the dimensionality is low. Although both methods achieve a slightly lower peak memory usage than DoRA, their memory footprint remains substantially higher than that of LoRA.

6 Conclusions

In this work, we examined some of the theoretical foundations of orthogonal fine-tuning. Based on our findings we proposed HOFT, a new PEFT method that adapts a pre-trained weight matrix by pre- and post-multiplying it with learned orthogonal matrices. We also developed SHOFT, a HOFT variant that introduces scaling transformations to further improve performance. Both exhibit good theoretical properties and provide higher flexibility. Our experimental results show that HOFT and SHOFT consistently match or outperform leading PEFT approaches across a wide range of benchmarks. To the best of our knowledge, QHOFT and QSHOFT are the first quantized orthogonal fine-tuning methods that maintain the benefits of their non-quantized counterparts, while operating with substantially reduced time and memory requirements.

For future work, we would like to extend our evaluation to include visual instruction tuning and the adaptation of multi-modal pre-trained models. In addition, we plan to explore how to reduce the number of trainable parameters in both methods, for instance by adopting vector-bank strategies similar to VB-LoRA. Finally, as discussed in Section 5, we would like to develop a variant of HOFT optimized for smaller weight matrices, aiming to reduce memory overhead and enforce orthogonality constraints.

References

[1] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

[2] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online, August 2021. Association for Computational Linguistics.

[3] Kai Biegun, Rares Dolga, Jake Cunningham, and David Barber. Rotrnn: Modelling long sequences with rotations. *arXiv preprint arXiv:2407.07239*, 2024.

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

[5] Thibaut Boissin, Franck Mamalet, Thomas Fel, Agustin Martin Picard, Thomas Massena, and Mathieu Serrurier. An adaptive orthogonal convolution scheme for efficient and flexible cnn architectures. *arXiv preprint arXiv:2501.07930*, 2025.

[6] E. Oran Brigham. *The Fast Fourier Transform and Its Applications*. Prentice Hall, Englewood Cliffs, NJ, 2 edition, 1974.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[8] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of NAACL-HLT*, pages 2924–2936, 2019.

[9] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[10] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[11] Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

[12] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *NeurIPS*, 2024.

[13] Yanhong Fei, Yingjie Liu, Xian Wei, and Mingsong Chen. O-vit: Orthogonal vision transformer. *arXiv preprint arXiv:2201.12133*, 2022.

[14] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 4th edition, 2013.

[15] John C. Gower and Garmt B. Dijksterhuis. *Procrustes Problems*, volume 30 of *Oxford Statistical Science Series*. Oxford University Press, Oxford, UK, 2004.

[16] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[17] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021.

[18] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

[19] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[20] Nam Hyeon-Woo, Moon Ye-Bin, and Tae-Hyun Oh. Fedpara: Low-rank hadamard product for communication-efficient federated learning. In *International Conference on Learning Representations*, 2022.

[21] Thierry Joffrain, Tze Meng Low, Enrique S. Quintana-Ortí, Robert van de Geijn, and Field G. Van Zee. Accumulating householder transformations, revisited. *ACM Trans. Math. Softw.*, 32(2):169–179, June 2006.

[22] Kshitij Khare and Christopher V. Stewart. Random Orthogonal Matrices and the Cayley Transform. *Bernoulli*, 26(2):807–825, 2019.

[23] Dawid J. Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. Vera: Vector-based random matrix adaptation. *ICLR*, 2024.

[24] Yang Li, Shaobo Han, and Shihao Ji. Vb-lora: Extreme parameter efficient fine-tuning with vector banks. *NeurIPS*, 2024.

[25] Valerii Likhosherstov, Jared Davis, Krzysztof Choromanski, and Adrian Weller. Cwy parametrization: a solution for parallelized optimization of orthogonal and stiefel matrices. In *International Conference on Artificial Intelligence and Statistics*, pages 55–63. PMLR, 2021.

[26] Rongmei Lin, Weiyang Liu, Zhen Liu, Chen Feng, Zhiding Yu, James M Rehg, Li Xiong, and Le Song. Regularizing neural networks via minimizing hyperspherical energy. *Conference on Computer Vision and Pattern Recognition*, 2020.

[27] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024.

[28] Weiyang Liu, Rongmei Lin, Zhen Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song. Learning towards minimum hyperspherical energy. *NeurIPS*, 2018.

[29] Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, et al. Parameter-efficient orthogonal finetuning via butterfly factorization. *International Conference on Learning Representations*, 2024.

[30] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37:121038–121072, 2024.

[31] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, 2018.

[32] Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. Orca-math: Unlocking the potential of slms in grade school math. *arXiv preprint arXiv:2402.14830*, 2024.

[33] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

[34] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2023.

[35] Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[36] Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. *Advances in Neural Information Processing Systems*, 36:79320–79362, 2023.

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[38] Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In Philipp Koehn, Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Tom Kocmi, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Marco Turchi, and Marcos Zampieri, editors, *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.

[39] Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November 2020. Association for Computational Linguistics.

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[41] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.

[42] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[43] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *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, 2019.

[44] Yang Sui, Miao Yin, Yu Gong, Jinqi Xiao, Huy Phan, and Bo Yuan. Elrt: Efficient low-rank training for compact convolutional neural networks. *arXiv preprint arXiv:2401.10341*, 2024.

[45] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[46] Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. Dylora: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3274–3287, 2023.

[47] Changhan Wang, Anne Wu, and Juan Pino. Covost 2 and massively multilingual speech-to-text translation. *arXiv preprint arXiv:2007.10310*, 2020.

[48] Changhao Wu, Shenan Zhang, Fangsong Long, Ziliang Yin, and Tuo Leng. Towards better orthogonality regularization with disentangled norm in training deep cnns. *arXiv preprint arXiv:2306.09939*, 2023.

[49] Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language models. In *ICLR*, 2024.

[50] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[51] Shih-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard BW Yang, Giyeong Oh, and Yanmin Gong. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. In *The Twelfth International Conference on Learning Representations*, 2023.

[52] Longhui Yu, Weisen Jiang, Han Shi, YU Jincheng, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*, 2023.

[53] Shen Yuan, Haotian Liu, and Hongteng Xu. Bridging the gap between low-rank and orthogonal adaptation via householder reflection adaptation. *NeurIPS*, 2024.

[54] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, 2019.

[55] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *11th International Conference on Learning Representations, ICLR 2023*, 2023.

[56] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2018)*, pages 586–595, Salt Lake City, Utah, USA, December 2018. IEEE Computer Society.

A Experimental details

A.1 Commonsense reasoning experiments

For commonsense reasoning experiments, we employ a NVIDIA A40 GPU for training LLaMA3.1-8B and Qwen2.5-7B models. For training Phi4-14B and Qwen2.5-14B, a NVIDIA H100 GPU was employed. For all experiments, the rank r was set to 16, and a dropout of 0.05. The optimizer employed was AdamW with Linear LR Scheduler. All models were trained for 2 epochs using a batch size of 4 and accumulation step of 4. The number of warmup steps was set to 100. The adapted layers were Query, Key, Value, Up, Down and Gate. We provide in Table 7 the learning rates used per model and per PEFT method.

Table 7: Learning rate hyperparameter configurations for commonsense reasoning experiments

Method	LLaMA3.1-8B	Qwen2.5-7B	Phi4-14B	Qwen2.5-14B
LoRA	9e-5	1e-4	9e-5	1e-4
DoRA	1e-4	9e-5	9e-5	9e-5
HOFT	1e-4	9e-5	9e-5	9e-5
SHOFT	2e-4	1e-4	9e-5	2e-4

A.2 Machine translation experiments

For machine translation experiments, we use a NVIDIA A30 GPU for training NLLB-3.3B model. For training LLaMA2-7B and LLaMA3.1-8B, a NVIDIA A40 GPU was used. For all experiments, the rank r was set to 16, and a dropout of 0.05. The optimizer employed was AdamW with Linear LR Scheduler. For French and German datasets, models were trained for 2 epochs on the first 10K elements of the training dataset. For Slovene and Latvian, models were trained for 3 epochs. All experiments use batch size of 16 and accumulation step of 4. The number of warmup steps was set to 100. The adapted layers were Query, Key and Value. We provide in Table 8 the learning rates used per language, model and per PEFT method.

Table 8: Learning rate hyperparameter configurations for machine translation experiments

Language	Method	NLLB-3.3B	LLaMA2-7B	LLaMA3.1-8B
Slovene	LoRA	4e-4	4e-4	8e-4
	DoRA	4e-4	4e-4	9e-4
	HOFT	5e-4	6e-4	1e-3
	SHOFT	5e-4	6e-4	7e-4
German	LoRA	6e-4	3e-4	4e-4
	DoRA	6e-4	3e-4	4e-4
	HOFT	2e-4	2e-4	8e-4
	SHOFT	2e-4	2e-4	4e-4
French	LoRA	5e-4	1e-4	1e-4
	DoRA	4e-4	1e-4	1e-4
	HOFT	1e-4	1e-4	3e-4
	SHOFT	4e-4	1e-4	1e-4
Latvian	LoRA	5e-4	4e-4	5e-4
	DoRA	5e-4	5e-4	5e-4
	HOFT	2e-4	9e-4	6e-4
	SHOFT	3e-4	8e-4	5e-4

A.3 Subject-driven generation experiments

For quantitative subject-driven experiments, we employ 10 NVIDIA A40 GPUs for training the Stable Diffusion 1.5 model. For all experiments, no dropout was used. The optimizer employed was AdamW with Linear LR Scheduler. All models were trained for 2005 steps using a batch size of 1. The adapted layers were Query, Key, Value and Out from the U-Net part. The learning rate used for training both HOFT and SHOFT is 5e-4.

For qualitative subject-driven experiments, we employ 5 NVIDIA A40 GPUs for training the Stable Diffusion XL model. For all experiments, no dropout was used. For all PEFT methods the rank r was set to 16, except for HRA, which was set to 32 for fair comparison. The optimizer employed was AdamW with Linear LR Scheduler. All models were trained 1000 steps using a batch size of 4 and gradient accumulation of 4. The adapted layers were Query, Key, Value and Out from the U-Net and text encoder part. The learning rate used for training both all PEFT methods is 1e-4.

A.4 Mathematical reasoning experiments

For mathematical reasoning experiments, we employ a NVIDIA H100 GPU for training LLaMA2-7B model. For all experiments, the rank r was set to 8, and no dropout. The optimizer employed was AdamW with Linear LR Scheduler. All models were trained for 2 epochs using a batch size of 8 and accumulation step of 2. The warmup ratio was set to 0.05. The adapted layers were Query and Value. The learning rates used by HOFT and SHOFT were 1e-3 and 7e-4 respectively.

A.5 Mathematical reasoning experiments with quantized models

For experiments in mathematical reasoning with quantized models, we employ a NVIDIA H100 GPU for training LLaMA2-7B and LLaMA3.1-8B models. Models are quantized using NF4 and double quatization. For all experiments, the rank r was set to 16, and a dropout of 0.05. The optimizer employed was AdamW with Linear LR Scheduler. For Orca-Math dataset, all models were trained for 2 epochs using a batch size of 4 and accumulation step of 1. For GSM8K dataset, all models were trained for 3 epochs using a batch size of 4 and accumulation step of 1. The number of warmup steps was set to 100. The adapted layers were Query, Key and Value. We provide in Table 9 the learning rates used per model and per PEFT method.

Table 9: Learning rate hyperparameter configurations for mathematical reasoning on quantized models

Dataset	Method	GSM8K	Orca-Math
LLaMA2-7B	QLoRA	4e-4	1e-4
	QDoRA	4e-4	9e-5
	QHOFT	3e-4	1e-4
	QSHOFT	3e-4	4e-4
LLaMA3.1-8B	QLoRA	2e-4	9e-5
	QDoRA	1e-4	9e-5
	QHOFT	3e-4	2e-4
	QSHOFT	4e-4	2e-4

B About inverse approximation error

B.1 Hyperspherical energy difference

Given $\mathbf{W} = (\mathbf{w}_1 \mid \cdots \mid \mathbf{w}_n) \in \mathbb{R}^{m\times n}$, where $\mathbf{w}_i$ denotes the i-th column of matrix $\mathbf{W}$, the hyperspherical energy is defined as follows:

$$\text{HE}(\mathbf{W}) = \sum_{i\neq j} \|\mathbf{w}_i - \mathbf{w}_j\|^{-1} \tag{7}$$

In order to measure the difference on the hyperspherical energy, we conduct an experiment by approximating two random gaussian accumulated householder transformations $\mathbf{Q_U}, \mathbf{Q_V}$. We measure $|\text{HE}(\mathbf{M}) - \text{HE}(\mathbf{Q_U}\mathbf{M}\mathbf{Q_V})|$, where $\mathbf{M}$ is a random gaussian matrix. Results are show in Figure 5.

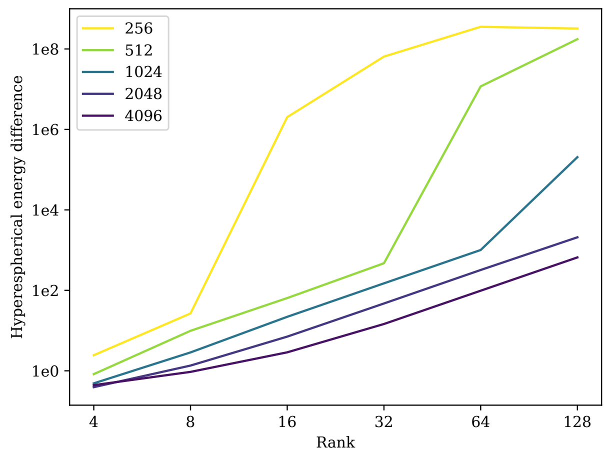


Figure 5: Hyperspherical energy difference

As observed in Figure 5, the hyperspherical energy tends to increase rapidly for higher ranks. Remarkably, for all cases the difference is negligible when $r = 1$ and $r = 2$ (omitted in Figure 5 for clarity). We can conclude from Figures 1 and 5 that, for a given rank r, the inverse approximation improves as the dimension of the matrix increases. Given the growing tendency for weight matrices in new pre-trained models, this is really convenient.

B.2 Indifference towards weight decay

One theoretical property of computing the CWY transform along with the inverse approximation is that, after applying weight decay to the original weights $\mathbf{U} \in \mathbb{R}^{m\times r}$, the resulting accumulated householder matrix remains the same. That is, given $\mathbf{U}' = \mathbf{U} - \lambda\mathbf{U}$, we compute

$$\begin{aligned}
\mathbf{Q}_{\mathbf{U}'} &= \mathbf{I} + \mathbf{U}' \left(\mathbf{D}'^{-1}\mathbf{A}'\mathbf{D}'^{-1} - \mathbf{D}'^{-1}\right) \mathbf{U}'^{\top} \\
&= \mathbf{I} + (1-\lambda)^2\mathbf{U} \left(\frac{1}{(1-\lambda)^2}\mathbf{D}^{-1}(1-\lambda)^2\mathbf{A}\frac{1}{(1-\lambda)^2}\mathbf{D}^{-1} - \frac{1}{(1-\lambda)^2}\mathbf{D}^{-1}\right) \mathbf{U}^{\top} \\
&= \mathbf{I} + \frac{(1-\lambda)^2}{(1-\lambda)^2}\mathbf{U}\left(\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1} - \mathbf{D}^{-1}\right)\mathbf{U}^{\top} \\
&= \mathbf{I} + \mathbf{U}\left(\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1} - \mathbf{D}^{-1}\right)\mathbf{U}^{\top} = \mathbf{Q_U}
\end{aligned}$$

Thus, we ensure that distance-preserving transformations in HOFT and SHOFT are not affected by weight decay. From this fact, we can ignore weight decay when adapting with HOFT. Additionally, when adapting with SHOFT, weight decay only affects the scaling transformation $\mathbf{m}$.

C Proof for Equation 1

Given $\mathbf{M}, \widehat{\mathbf{M}} \in \mathbb{R}^{m\times n}$ two matrices such that both have the same hyperspherical energy. Then

$$\begin{aligned}\min_{\mathbf{Q}\in \mathrm{O}(m)} \left\|\widehat{\mathbf{M}} - \mathbf{Q}\mathbf{M}\right\|_F &= \min_{\mathbf{Q}\in \mathrm{O}(m)} \left\|\widehat{\mathbf{Q}}_{\mathbf{U}}\mathbf{M}\widehat{\mathbf{Q}}_{\mathbf{V}} - \mathbf{Q}\mathbf{M}\right\|_F \leq \left\|\mathbf{M}\widehat{\mathbf{Q}}_{\mathbf{V}} - \mathbf{M}\right\|_F \\ &= \left\|\mathbf{M}\left(\widehat{\mathbf{Q}}_{\mathbf{V}} - \mathbf{I}\right)\right\|_F \leq \|\mathbf{M}\|_F \left\|\widehat{\mathbf{Q}}_{\mathbf{V}} - \mathbf{I}\right\|_F\end{aligned}$$

Now we need to compute $\left\|\widehat{\mathbf{Q}}_{\mathbf{V}} - \mathbf{I}\right\|_F$

$$\begin{aligned}\left\|\widehat{\mathbf{Q}}_{\mathbf{V}} - \mathbf{I}\right\|_F^2 &= \mathrm{Tr}\left(\left(\widehat{\mathbf{Q}}_{\mathbf{V}} - \mathbf{I}\right)^\top \left(\widehat{\mathbf{Q}}_{\mathbf{V}} - \mathbf{I}\right)\right) = \mathrm{Tr}\left(\widehat{\mathbf{Q}}_{\mathbf{V}}^\top\widehat{\mathbf{Q}}_{\mathbf{V}} - \widehat{\mathbf{Q}}_{\mathbf{V}}^\top - \widehat{\mathbf{Q}}_{\mathbf{V}} + \mathbf{I}\right) = \\ &= 2m - \mathrm{Tr}\left(\widehat{\mathbf{Q}}_{\mathbf{V}}^\top\right) - \mathrm{Tr}\left(\widehat{\mathbf{Q}}_{\mathbf{V}}\right) = 2m - 2\mathrm{Tr}\left(\widehat{\mathbf{Q}}_{\mathbf{V}}\right)\end{aligned}$$

The previous expression attains its maximum precisely when $\widehat{\mathbf{Q}}_{\mathbf{V}} = -\mathbf{I}$. In that case, we conclude that $\mathrm{Tr}\left(\widehat{\mathbf{Q}}_{\mathbf{V}}\right) = -m$ and consequently $\left\|\widehat{\mathbf{Q}}_{\mathbf{V}} - \mathbf{I}\right\|_F^2 \leq 4m$. Thus, final upper-bound will be

$$\min_{\mathbf{Q}\in \mathrm{O}(m)} \left\|\widehat{\mathbf{M}} - \mathbf{Q}\mathbf{M}\right\|_F \leq 2\sqrt{m}\,\|\mathbf{M}\|_F$$

D Time and memory consumption

In order to give a better understanding of the time and memory complexity of HOFT and SHOFT, we provide the runtime for training and the peak memory usage during training from the commonsense reasoning, qualitative subject-driven generation and mathematical reasoning using quantized models experiments. All values are gathered in Tables 10, 11 and 12.

In Table 10 we can observe that both HOFT and SHOFT are 72.5% and 55.3% faster on average than DoRA, respectively. With respect to LoRA, they are on average 35.1% and 41.8% slower, respectively. In terms of memory, both HOFT and SHOFT peak memories are between LoRA's and DoRA's peak memories, except in Phi4-14B, where the memory is higher in HOFT and SHOFT. This unusual peak in Phi4-14B is due to the fact Query, Key and Value are all together in a matrix (the same happens with Up and Gate layers).

Table 10: Memory and time complexity comparison on the commonsense reasoning task

Model	**Method**	**Training time (hours)**	**Peak memory (GB)**
LLaMA3.1-8B	LoRA	1.5	31.9
	DoRA	3.3	45.8
	HOFT	2.3	42.3
	SHOFT	2.6	44.5
Qwen2.5-7B	LoRA	6.1	30.7
	DoRA	13.9	44.5
	HOFT	8.0	41.3
	SHOFT	9.2	43.5
Phi4-14B	LoRA	2.4	49.9
	DoRA	8.3	68.0
	HOFT	3.4	78.6
	SHOFT	3.7	78.0
Qwen2.5-14B	LoRA	2.8	39.8
	DoRA	7.6	59.4
	HOFT	5.9	56.4
	SHOFT	6.4	57.6

From Table 11, both HOFT and SHOFT are 72.1% faster than OFT, and 732.5% faster than HRA. In this respect, it is worth noting that HRA entails a number of sequential householder transformations that leads to a comparatively high training time. With respect to LoRA, they are on average 32.6% slower. In terms of memory, both HOFT and SHOFT require less memory than OFT and HRA, and the same as LoRA.

Table 11: Memory and time complexity comparison on the mathematical reasoning using quantized models experiments

Method	Training time (hours)	Peak memory (GB)
LoRA	2.9	25.5
HRA	35.8	25.7
OFT	7.4	26.7
HOFT	4.3	25.5
SHOFT	4.3	25.5

Table 12 shows that there is a minor difference in time cost between LoRA, HOFT and SHOFT. In the case of DoRA, it is 16.7% slower than the rest. In terms of memory, both HOFT and SHOFT peak memories are between LoRA's and DoRA's peak memories, requiring at most 9.6% and 20.8% more memory than LoRA, respectively.

Table 12: Memory and time complexity comparison on the mathematical reasoning using quantized models experiments

Model	Method	Training time (hours)	Peak memory (GB)
LLaMA2-7B	QLoRA	0.9	43.2
	QDoRA	1.2	58.2
	QHOFT	1.0	47.7
	QSHOFT	1.0	52.2
LLaMA3.1-8B	QLoRA	1.0	52.0
	QDoRA	1.2	60.4
	QHOFT	1.0	57.0
	QSHOFT	1.0	56.4

E Additional experiments

E.1 Rank exploration

We explore the effect of various rank settings $r \in \{2, 4, 8, 16, 32, 64\}$ on LoRA, DoRA, HOFT and SHOFT by evaluating the fine-tuned LLaMA3.1-8B and Qwen2.5-7B performance on the commonsense reasoning tasks described in Section 4.1. The implementation settings are the same as those for rank 16, given in Appendix A.

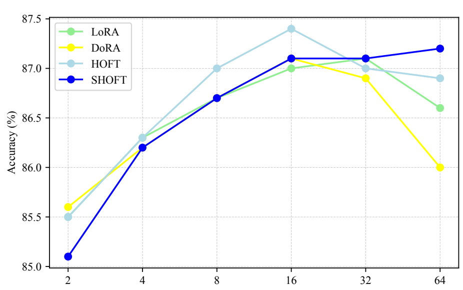


Figure 6: Rank exploration in LLaMA3.1-8B

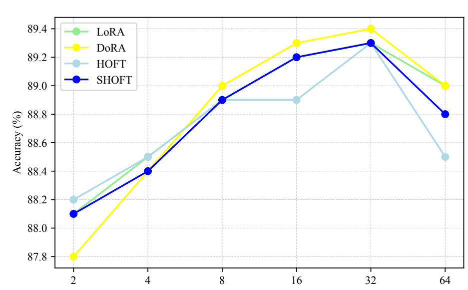


Figure 7: Rank exploration in Qwen2.5-7B

The average accuracies of the PEFT methods across different ranks are shown in Figures 6 and 7. In Figure 6, all four methods improve sharply up to $r = 16$, where HOFT peaks. Beyond $r = 16$, DoRA's performance declines markedly while LoRA falls slightly. In contrast, SHOFT maintains a mild upward trend approaching HOFT best result for $r = 64$. In Figure 7, the methods again climb to a peak at $r = 32$, where DoRA attains the highest accuracy, with SHOFT and LoRA close behind. At $r = 64$, HOFT's accuracy falls more noticeably, whereas the others dip only slightly.

Overall, these results suggest that HOFT is the strongest option for moderate ranks, but SHOFT is the most robust method at higher ranks and offers the steadiest, most consistent gains across the entire rank spectrum.

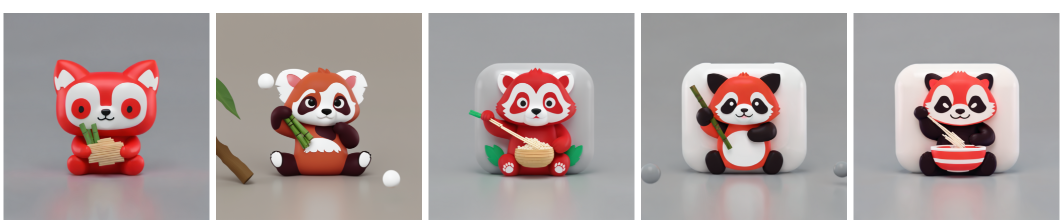
E.2 More qualitative results on subject-driven generation

Prompt: a TOK pink 3d icon of a rainbow unicorn eating marshmallow, in the style of TOK

Prompt: a TOK 3d icon of a yellow racoon eating banana, in the style of TOK

Prompt: a TOK 3d icon of a yellow duck eating sushi, in the style of TOK

Prompt: a TOK 3d icon of a demon red panda eating bamboo, in the style of TOK



LoRA | HRA | OFT | HOFT | SHOFT

Figure 8: Comparison of different prompts in 3D icons dataset

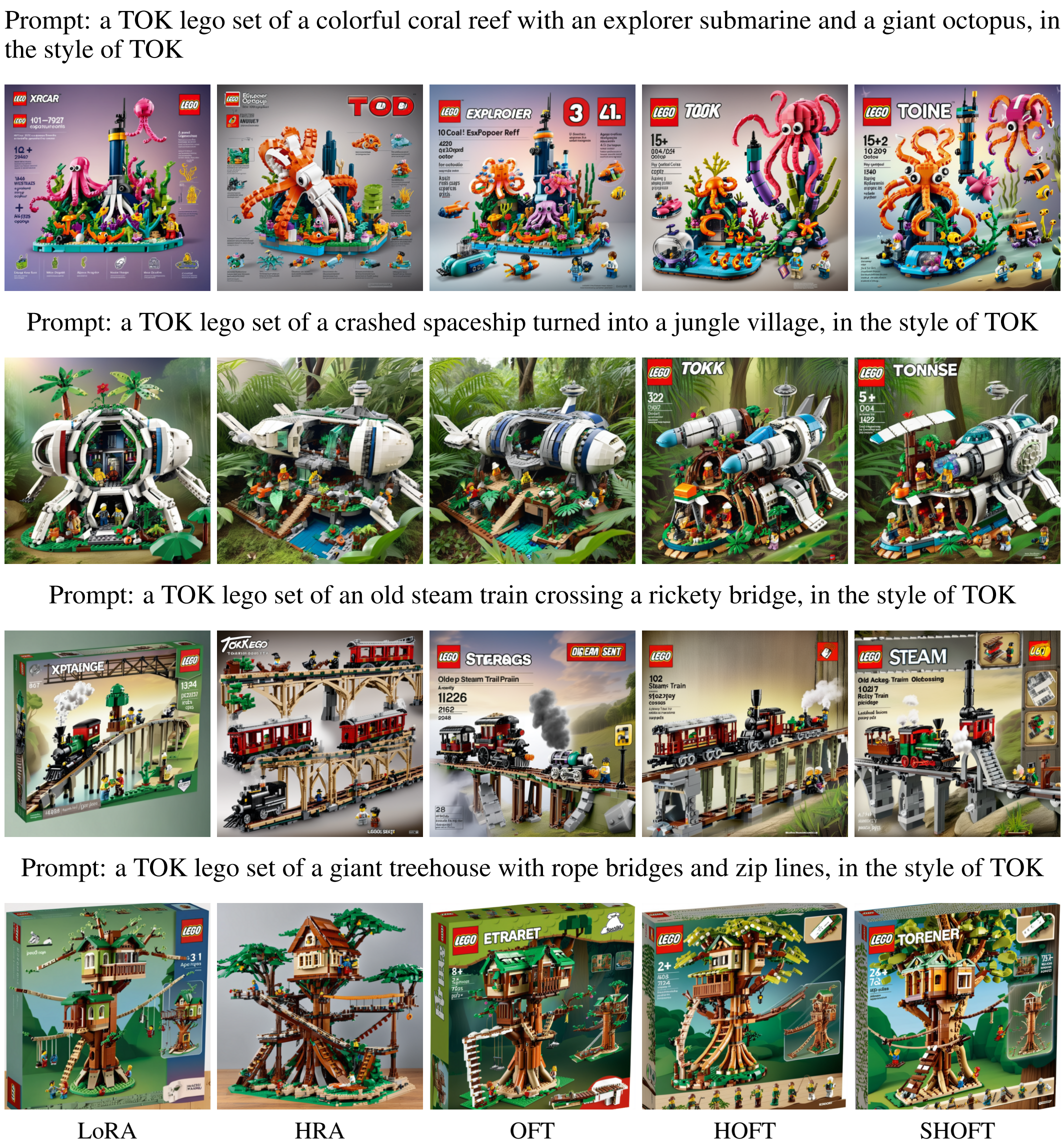


Figure 9: Comparison of different prompts in lego sets dataset

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The key assertions in the abstract and introduction faithfully capture the paper's contributions and overall scope.

   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 discussed the limitations of our work in Section 5.

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: Each theoretical result is accompanied by a complete list of assumptions and a rigorous, fully detailed proof.

   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: For each experiment, we specify the hardware used, the model architecture and all required hyperparameters.

   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: [Yes]

   Justification: All code is included in the supplementary material, complete with a structural overview, and the datasets are publicly available online with easy download instructions provided there.

   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: The appendix includes comprehensive experimental details, such as hyperparameter settings, optimizer configurations, and more.

   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 omitted error bars due to the large number of experiments. The significant time requirements that reporting them would entail are not practical.

   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: For each experiment, we provide the information of the GPU used. In some experiments, we also include train runtime and GPU peak memory usage. All that information is on the appendix.

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: This paper adheres in every respect to the NeurIPS Code of Ethics.

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: [Yes]

Justification: HOFT reduce the computational and memory overhead of fine-tuning large language models and opens the door to a broader range of applications. Lowering both cost and technical barriers increases accessibility carries inherent risks: like other PEFT techniques, HOFT may be used to reinforce or propagate harmful biases present in training data, generate misleading or malicious content, or facilitate inappropriate applications when oversight is lacking. Crucially, these challenges are not unique to HOFT but reflect broader issues in the development and deployment of fine-tuning methods. Addressing these areas represents an important avenue for future research as we work to ensure that HOFT (and PEFT techniques more generally) are harnessed safely and equitably.

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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    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: We have appropriately acknowledged all creators and original owners of the code, data, and models used in this work.

    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: Comments are provided in the code of the supplementary material.

    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: This paper does not involve crowdsourcing nor research with human subjects.

    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: This paper does not involve crowdsourcing nor research with human subjects.

    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: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    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.