KQ-SVD: Compressing the KV Cache with Provable Guarantees on Attention Fidelity

Damien Lesens
ENS de Lyon

Beheshteh T. Rakhshan
DIRO, Université de Montréal
Mila

Guillaume Rabusseau
DIRO, Université de Montréal
Mila - CIFAR AI Chair

Abstract

The Key-Value (KV) cache is central to the efficiency of transformer-based large language models (LLMs), storing previously computed vectors to accelerate inference. Yet, as sequence length and batch size grow, the cache becomes a major memory bottleneck. Prior compression methods typically apply low-rank decomposition to keys alone or attempt to jointly embed queries and keys, but both approaches neglect that attention fundamentally depends on their inner products. In this work, we prove that such strategies are sub-optimal for approximating the attention matrix. We introduce KQ-SVD, a simple and computationally efficient method that directly performs an optimal low-rank decomposition of the attention matrix via a closed-form solution. By targeting the true source of redundancy, KQ-SVD preserves attention outputs with higher fidelity under compression. Extensive evaluations on LLaMA and Mistral models demonstrate that our approach consistently delivers superior projection quality.

1 INTRODUCTION

The rise of Large Language Models (LLMs) [Touvron et al., 2023, Chaplot, 2023, Achiam et al., 2023, Guo et al., 2025] has expanded AI capabilities beyond earlier models. Transformers [Vaswani et al., 2017] replace recurrence with self-attention, enabling parallelism and improved sequence modeling, but their quadratic memory and computation costs limit long-sequence scalability.

Key-Value (KV) caches are introduced to accelerate autoregressive generation by storing intermediate attention KV vectors, avoiding redundant computation of shared prefixes for each generated token. Although KV caching reduces computational overhead, it substantially increases memory consumption, as the cache size grows linearly with both sequence length and batch size. This trade-off motivates the development of KV cache compression techniques, which are crucial for enabling efficient and cost-effective deployment of LLMs across diverse hardware platforms [Fu, 2024, Shi et al., 2024]. Variants like Multi-Query Attention (MQA) [Shazeer, 2019] and Grouped-Query Attention (GQA) [Ainslie et al., 2023] reduce KV cache size by sharing or grouping query vectors while maintaining performance comparable to full Multi-Head Attention (MHA). However, they may introduce accuracy trade-offs and hardware sensitivity, which can affect performance generalization. Additional approaches, including sparse [Zhang et al., 2021] and linearized [Katharopoulos et al., 2020] attention, further reduce computational and memory costs, shaping KV cache optimization strategies.

Another promising line of research exploits the low-rank structure of KV caches to reduce memory overhead. Multi-Head Latent Attention (MLA) [Liu et al., 2024, Guo et al., 2025] maps tokens into the low-rank latent space and stores these compressed representations in place of the original key and value states. However, using MLA necessitates training the model from the ground up. In contrast, ASVD [Yuan et al., 2024], LoRC [Zhang et al., 2024], and Palue [Chang et al., 2025b] apply SVD to key-value parameter matrices without retraining to build low-rank projection modules. A key limita-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

tion of these approaches is that they often compress only the keys, neglecting the query-key interaction that underlies attention. EigenAttention [Saxena et al., 2024] and Zack [Zhang and Shen, 2024] attempt to address this by incorporating both queries and keys in low-rank decompositions, yet their behavior largely resembles that of SVD-based methods that compress keys alone.

In this work, we address these limitations by introducing KQ-SVD, a compression method that achieves optimal low-rank approximation of the attention matrix efficiently and in closed form. Our method explicitly captures the interactions between queries and keys through their inner products, preserving the fundamental structure of attention. Beyond key-query interactions, we also consider the corresponding interactions between values and the output projection, enhancing the fidelity of the approximation. By leveraging the inherent low-rank structure of KV caches [Yu et al., 2024, Saxena et al., 2024], we formulate attention matrix approximation as a principled low-rank decomposition problem. Our theoretical analysis quantifies the error between prior key-only SVD approaches and our optimal method, and shows that methods incorporating both queries and keys can degrade when keys and queries are rescaled by the same factor, effectively behaving like key-only SVD methods. Our contributions can be summarized as follows:

- We introduce KQ-SVD, an optimal low-rank approximation of the attention matrix capturing key-query interactions.
- We theoretically quantify the advantages of KQ-SVD over methods based on key low-rank decomposition and SVD on concatenated queries and keys.
- We show that KQ-SVD is compatible with and also optimal in the Grouped-Query Attention setting.
- We provide extensive empirical evaluations with LLaMA2-7B, LLaMA2-13B, LLama3-8B and Mistral-7B models on the C4 dataset demonstrating significant advantages of KQ-SVD over existing low rank projection methods.

2 RELATED WORKS

Low-rank structure of the KV-cache. Several methods exploit the inherent low-rank structure of cached key-value (KV) matrices to reduce memory footprint. ECKVH [Yu et al., 2024] compresses the cache by grouping attention heads, performing singular value decomposition (SVD) within each group, and retaining only the dominant singular components. EigenAttention [Saxena et al., 2024] generalizes this idea by constructing low-rank bases that jointly approximate queries, keys, and values, effectively lowering the dimensionality of KV representations. Q-Filters [Godey et al.] introduces a training-free variant, projecting keys into a low-rank subspace via SVD to approximate attention scores efficiently with minimal accuracy loss. Moreover, [Yu et al., 2024] investigates the intrinsic low-rank nature of KV caches and compresses KV heads through careful grouping and SVD-based decomposition. In contrast, Loki [Singhania et al., 2024] adopts a two-stage strategy: it first estimates approximate attention scores in a lower-dimensional space to rank and select the most relevant keys, and then computes exact attention scores using only the selected keys, reducing both memory and computational cost.

KV Weights Compression. An alternative approach targets the KV weight matrices themselves rather than the cached matrices. LoRC [Zhang et al., 2024] applies low-rank approximations directly to the key and value weight matrices, achieving compression at the parameter level. Palu [Chang et al., 2025b] follows a similar strategy, jointly compressing key and value weight matrices via SVD. ShadowKV [Sun et al., 2024] introduces a distinct perspective by performing SVD on pre-RoPE key matrices to reduce their dimensionality, demonstrating the versatility of low-rank methods in optimizing KV representations.

Positioning of KQ-SVD. Although prior methods have made significant strides in compressing KV caches and their weight matrices, they often treat keys and values independently or approximate attention only indirectly, leaving the core query-key interactions underrepresented and leading to sub-optimal low rank approximation of attention matrices. KQ-SVD addresses these limitations by formulating a principled, closed-form low-rank approximation of the full attention matrix.

3 PRELIMINARY

In this section, we introduce our notations and present the necessary background on Multi-Head Attention (MHA).

3.1 Notations

We use lower case bold letters for vectors (e.g., $\mathbf{a}, \mathbf{b}$), upper case bold letters for matrices (e.g., $\mathbf{A}, \mathbf{B}$). $\mathbf{A}^+$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{A}$. Throughout the paper, the singular value decomposition (SVD) of a matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$ is presented by $\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, with $\mathbf{U} \in \mathbb{R}^{m\times n}$, $\mathbf{V} \in \mathbb{R}^{n\times n}$ matrices with orthonormal columns and $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ a diagonal matrix with positive diagonal entries $\{\sigma_i\}_{i=1}^n$. The columns of $\mathbf{U}$ and $\mathbf{V}$ are called respectively the left and right singular vectors of $\mathbf{S}$, and the σ_i's the singular values of $\mathbf{S}$, noted $\sigma_i(\mathbf{S})$. The optimal rank-R approximation of $\mathbf{S}$ with respect to the Frobenius norm can be obtained via the SVD by truncating it to keep only the first R singular vectors and singular values: $\mathbf{S} \approx \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^\top$ with $\hat{\mathbf{U}} \in \mathbb{R}^{m\times R}$, $\hat{\mathbf{V}} \in \mathbb{R}^{n\times R}$ and $\hat{\mathbf{\Sigma}} \in \mathbb{R}^{R\times R}$. The column space of $\mathbf{S}$ is noted $\mathcal{R}(\mathbf{S})$.

3.2 Background

In transformer architectures, self-attention assigns relative importance to tokens, enabling the model to selectively focus on different segments of the input sequence. For a sequence of token embeddings $\mathbf{X} \in \mathbb{R}^{T\times D}$, multi-head attention is computed as

$$\mathrm{MHA}(\mathbf{X}) = [\mathbf{H}_1, \dots, \mathbf{H}_h]\mathbf{W}^O,$$

where

$$\mathbf{H}_i = \mathrm{Softmax}\Big(\frac{\mathbf{Q}_i\mathbf{K}_i^\top}{\sqrt{d}}\Big)\mathbf{V}_i,$$

with $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{D\times d}$, $d = D/h$, $\mathbf{Q}_i = \mathbf{X}\mathbf{W}_i^Q$, $\mathbf{K}_i = \mathbf{X}\mathbf{W}_i^K$, $\mathbf{V}_i = \mathbf{X}\mathbf{W}_i^V$, and $\mathbf{W}^O \in \mathbb{R}^{D\times D}$. In masked attention, the upper-diagonal entries of the attention matrix $\mathbf{Q}_i\mathbf{K}_i^\top$ are set to $-\infty$ to prevent a token from attending to future positions.

This computation scales quadratically with the sequence length T. In auto-regressive decoding, previously computed key and value vectors are cached to avoid redundant computation, reducing the per-token cost. Specifically at time T, for each head, new key-value pairs are concatenated to the caches $\mathbf{K}$ and $\mathbf{V}$, followed by an attention computation:

$$\mathbf{K} \leftarrow \mathrm{Concat}(\mathbf{K}, \mathbf{k}_T),\ \mathbf{V} \leftarrow \mathrm{Concat}(\mathbf{V}, \boldsymbol{v}_T),$$
$$\mathbf{h}_T = \mathrm{Softmax}\Big(\frac{\mathbf{q}_T\mathbf{K}^\top}{\sqrt{d}}\Big)\mathbf{V}$$

where $\mathbf{k}_T = \mathbf{x}_T\mathbf{W}^K$, $\boldsymbol{v}_T = \mathbf{x}_T\mathbf{W}^V$, $\mathbf{q}_T = \mathbf{x}_T\mathbf{W}^Q$. While caching mitigates redundant computation, generating the T-th token still incurs $\mathcal{O}(T)$ cost, and the memory footprint of stored keys and values grows linearly with sequence length. For sufficiently long contexts, this memory requirement becomes a dominant bottleneck, as the cumulative size of the KV cache can surpass the model parameters by several orders of magnitude.

In the following sections, we briefly review the K-SVD method [Chang et al., 2025b, Yu et al., 2024, Zhang et al., 2024], which compresses key representations using singular value decomposition (SVD), and the Eigen approach [Saxena et al., 2024], which jointly considers keys and queries by vertically concatenating them and applying SVD to the resulting matrix.

3.3 Cache compression with SVD

Recent works [Chang et al., 2025b, Zhang et al., 2024, Chang et al., 2025a] demonstrate that singular value decomposition (SVD) is a powerful tool for compressing the KV cache in large language models, as the cache exhibits low-rank structure. Let $\mathbf{K} = \mathbf{U}_K\mathbf{\Sigma}_K\mathbf{V}_K^\top \in \mathbb{R}^{T\times d}$ be the SVD of the key matrix, and let $\widetilde{\mathbf{K}} = \hat{\mathbf{U}}_K\hat{\mathbf{\Sigma}}_K\hat{\mathbf{V}}_K^\top$ denote its rank-R truncated version. By the Eckart-Young-Mirsky theorem, $\widetilde{\mathbf{K}}$ is the best rank-R approximation of $\mathbf{K}$ under the Frobenius norm. In other words, the optimization problem

$$\min_{\mathbf{P}\in\mathbb{R}^{d\times d}} \|\mathbf{K}\mathbf{P} - \mathbf{K}\|_F^2 \ \text{ s.t. } \mathrm{rank}(\mathbf{P}) \le R,$$

is solved by $\mathbf{P} = \hat{\mathbf{V}}_K\hat{\mathbf{V}}_K^\top$, leading to the approximation $\widetilde{\mathbf{K}} = \mathbf{K}\hat{\mathbf{V}}_K\hat{\mathbf{V}}_K^\top = \hat{\mathbf{U}}_K\hat{\mathbf{\Sigma}}_K\hat{\mathbf{V}}_K^\top$. Applying the same procedure to the value matrix $\mathbf{V} = \mathbf{U}_V\mathbf{\Sigma}_V\mathbf{V}_V^\top$, we can approximate the attention output as

$$\begin{aligned}\widetilde{\mathbf{H}} &= \mathrm{Softmax}(\mathbf{Q}\widetilde{\mathbf{K}}^\top/\sqrt{d})\widetilde{\mathbf{V}}\\ &= \mathrm{Softmax}(\mathbf{Q}\hat{\mathbf{V}}_K\hat{\mathbf{V}}_K^\top\mathbf{K}^\top/\sqrt{d})\mathbf{V}\hat{\mathbf{V}}_V\hat{\mathbf{V}}_V^\top.\end{aligned}$$

This formulation is particularly useful because it allows to store only the compressed caches $\mathbf{K}\hat{\mathbf{V}}_K$ and $\mathbf{V}\hat{\mathbf{V}}_V$ in memory. These matrices are of size $R \times T$ instead of $d\times T$, resulting in significant memory savings. At runtime, queries are multiplied by $\hat{\mathbf{V}}_K$, while $\hat{\mathbf{V}}_V^\top$ can be absorbed into the output projection $\mathbf{W}_O$, streamlining computation.

A key advantage of this approach is that the SVD does not need to be computed during token generation. Instead, it can be performed once in a post-training calibration phase. For each layer l and attention head i, we only need to determine a basis $\mathbf{V}_{i,l} \in \mathbb{R}^{d\times R}$ such that $\mathbf{K}_{i,l} \approx \mathbf{K}_{i,l}\mathbf{V}_{i,l}(\mathbf{V}_{i,l})^\top$ with an analogous construction for the values. This basis can be learned from a calibration set of sequences.

Specifically, we pass n_s calibration sequences (e.g., sampled from a high-quality dataset such as C4 [Raffel et al., 2020]) through the model. The kth sequence produces caches $\mathbf{K}_{i,l}^k$ and $\mathbf{V}_{i,l}^k$ for every layer l and head i. These are concatenated to form large cache matrices $\mathbf{K}_{i,l} = \left(\mathbf{K}_{i,l}^1, \mathbf{K}_{i,l}^2, \ldots, \mathbf{K}_{i,l}^{n_s}\right)$. This aggregated cache provides a representative sample of the key vectors that will appear during inference. Performing SVD on $\mathbf{K}_{i,l}$ then yields the dominant singular vectors, which form a suitable low-rank basis for compression. The cost of generating calibration caches and computing the SVDs is negligible compared to model training, and is offset by the runtime speedups from cache compression. In the following, we will refer to this method as K-SVD.

Rank selection. The compression rank R is determined per layer by examining the singular value spectrum. Let $\{\sigma_j\}_j$ denote the singular values of a matrix $\mathbf{M}$. For a relative error tolerance ϵ, we select the smallest R such that

$$\|\mathbf{M} - \widetilde{\mathbf{M}}\|_F^2 \leq \epsilon \|\mathbf{M}\|_F^2 \Leftrightarrow \frac{\sum_{j=1}^R \sigma_j^2}{\sum_{i=1}^d \sigma_j^2} \geq 1 - \epsilon.$$

Prior studies [Yu et al., 2024, Saxena et al., 2024] have shown that KV matrices are indeed approximately low-rank, so substantial compression can be achieved with small error budget ϵ. The chosen rank may differ for keys and values depending on their spectra.

3.4 Cache compression with Eigen

Other works [Saxena et al., 2024, Zhang and Shen, 2024] emphasize that queries should also be considered when compressing key caches. Indeed, by projecting the key cache we also project the query matrix: $\mathbf{Q}\hat{\mathbf{V}}_K\hat{\mathbf{V}}_K^\top\mathbf{K}^\top = (\mathbf{Q}\hat{\mathbf{V}}_K\hat{\mathbf{V}}_K^\top)(\mathbf{K}\hat{\mathbf{V}}_K\hat{\mathbf{V}}_K^\top)^\top$ as $\hat{\mathbf{V}}_K\hat{\mathbf{V}}_K^\top$ is an idempotent matrix. Hence, it makes sense to compute the low rank projection by solving

$$\min_{\mathbf{S}} \|\mathbf{K} - \mathbf{K}\mathbf{S}\|_F^2 + \|\mathbf{Q} - \mathbf{Q}\mathbf{S}\|_F^2 \text{ s.t. } \operatorname{rank}(\mathbf{S}) \leq R,$$

which is equivalent to

$$\min_{\mathbf{S}} \left\| \begin{bmatrix}\mathbf{K}\\ \mathbf{Q}\end{bmatrix} - \begin{bmatrix}\mathbf{K}\\ \mathbf{Q}\end{bmatrix}\mathbf{S} \right\|_F^2 \text{ s.t. } \operatorname{rank}(\mathbf{S}) \leq R,$$

so that $\mathbf{S}$ will approximate queries and keys simultaneously. As the second formulation of the optimization problem shows, $\mathbf{S}$ can be computed by performing an SVD on the combined matrix $\begin{bmatrix}\mathbf{K}\\ \mathbf{Q}\end{bmatrix}$. This approach ensures that the learned projection preserves keys and queries while reducing dimensionality. We refer to this approach as Eigen throughout this paper. The calibration process follows the same procedure as K-SVD: large calibration caches are formed by using a collection of calibration sequences. The only difference is that query matrices from the calibration set are also used in the projection computation.

4 METHODOLOGY

In this section, we introduce our proposed approach KQ-SVD for KV cache compression. We consider taking the interaction between queries and keys into account. The method views the key and query matrices as a single entity and applies singular value decomposition (SVD) to $\mathbf{K}\mathbf{Q}^\top$. We begin by outlining the motivation behind this idea, followed by a detailed explanation of the technique.

4.1 Motivation

Existing compression methods based on SVD typically compress keys or jointly embed $\mathbf{Q}/\mathbf{K}/\mathbf{V}$. Theorem 1 (proof in Appendix A) inspired by [Wang et al., 2025] shows why that can fail: perturbations in $\mathbf{K}$ are amplified by the inner products $\mathbf{Q}\mathbf{K}^\top$ and further propagated by the value multiplication.

Theorem 1. *Let $\mathbf{X} \in \mathbb{R}^{T\times d}$ be a sequence of token embeddings, $\mathbf{K}, \mathbf{Q}, \mathbf{V} \in \mathbb{R}^{T\times d}$ and*

$$\mathrm{MHA}(\mathbf{X}) = \left[\mathrm{Softmax}(\mathbf{Q}_i\mathbf{K}_i^\top/\sqrt{d})\mathbf{V}_i)\right]_i \mathbf{W}^O,$$

$$\widetilde{\mathrm{MHA}}(\mathbf{X}) = \left[\mathrm{Softmax}(\mathbf{Q}_i\widetilde{\mathbf{K}_i}^\top/\sqrt{d})\widetilde{\mathbf{V}}_i)\right]_i \mathbf{W}^O,$$

where $\widetilde{\mathrm{MHA}}(\mathbf{X}), \widetilde{\mathbf{K}}_i$ and $\widetilde{\mathbf{V}}_i$ represent the approximation of $\mathrm{MHA}(\mathbf{X}), \mathbf{K}_i$ and $\mathbf{V}_i$, respectively. The difference between the actual attention output and the one produced with approximate keys and values is upper bounded as

$$\begin{aligned}&\|\widetilde{\mathrm{MHA}}(\mathbf{X}) - \mathrm{MHA}(\mathbf{X})\|_2\\ &\leq \sum_{i=1}^h \frac{\|\mathbf{V}_i\mathbf{W}_i^O\|_2}{\sqrt{d}}\|\mathbf{Q}_i\mathbf{K}_i^\top - \mathbf{Q}_i\widetilde{\mathbf{K}_i}^\top\|_2\\ &+ \|\mathbf{V}_i\mathbf{W}_i^O - \widetilde{\mathbf{V}}_i\mathbf{W}_i^O\|_2. \end{aligned} \tag{1}$$

The method proposed by this paper stems from the will to minimize this upper bound on the output approximation error. In each attention head, for a key cache $\mathbf{K}$ and a calibration set of query vectors $\{\mathbf{q}^{(j)}\}_{j\in\text{calibration}}$ (row vectors of size d), we want to

minimize

$$\sum_{j \in \text{calibration}} \|\mathbf{q}^{(j)}\mathbf{K}^\top - \mathbf{q}^{(j)}\widetilde{\mathbf{K}}^\top\|_2^2,$$

over low-rank $\widetilde{\mathbf{K}}$. This objective is equivalent to a low-rank approximation of the query-key interaction matrix and admits an optimal closed-form solution via the singular value decomposition. Similarly, to compress values we directly optimize the second term of Eq. (3) over low-rank $\widetilde{\mathbf{V}}$ for each attention head (see Appendix B).

4.2 Proposed Method

In the following, we drop the head index i for ease of clarity. Our method addresses cache compression by producing an optimal solution to

$$\min_{\widetilde{\mathbf{K}}} \|\mathbf{Q}\mathbf{K}^\top - \mathbf{Q}\widetilde{\mathbf{K}}^\top\|_F^2 \text{ s.t. } \operatorname{rank}(\widetilde{\mathbf{K}}) \leq R,$$

with $\mathbf{Q} = \left(\mathbf{q}^{(j)}\right)_{j \in \text{calibration}}$ the collection of query vectors from the calibration set. This formulation jointly considers the properties of both $\mathbf{K}$ and $\mathbf{Q}$, while taking into account their interaction through the inner product. The optimization problem can be restated using a projection matrix as

$$\min_{\mathbf{S}} \|\mathbf{K}\mathbf{S}\mathbf{Q}^\top - \mathbf{K}\mathbf{Q}^\top\|_F^2 \text{ s.t. } \operatorname{rank}(\mathbf{S}) \leq R,$$

where $\mathbf{S} \in \mathbb{R}^{d \times d}$. To be able to compress the key cache during the attention computation, we write the projection matrix $\mathbf{S}$ as a product of two matrices, i.e., $\mathbf{S} = \mathbf{A}\mathbf{B}$ with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{d \times R}$. The minimization problem tackled by KQ-SVD is thus

$$\min_{\mathbf{A},\mathbf{B} \in \mathbb{R}^{d \times R}} \|\mathbf{K}\mathbf{A}\mathbf{B}^\top\mathbf{Q}^\top - \mathbf{K}\mathbf{Q}^\top\|_F^2. \tag{2}$$

Given $\mathbf{A}^*$ and $\mathbf{B}^*$ the solutions of this optimization problem, we store the low-rank matrix $\mathbf{K}\mathbf{A}^* \in \mathbb{R}^{T \times R}$, which allows for compression while maintaining an accurate reconstruction of the attention matrix. The same strategy also applies to the value-output matrices (see Appendix B). In the next section, we state the main theorem which demonstrates that the optimization problem described above admits an optimal solution which can be computed efficiently.

4.3 KQ-SVD: Optimal attention factorization

The following theorem establishes the provably optimal low-rank factorization of the key-query matrix, which admits a simple closed-form solution.

Theorem 2. *Let $\mathbf{K}, \mathbf{Q} \in \mathbb{R}^{T \times d}$ be key and query cache matrices. The optimal solution to the low rank attention approximation problem*

$$\min_{\mathbf{A},\mathbf{B} \in \mathbb{R}^{d \times R}} \|\mathbf{K}\mathbf{A}\mathbf{B}^\top\mathbf{Q}^\top - \mathbf{K}\mathbf{Q}^\top\|_F,$$

is given by

$$\mathbf{A} = \mathbf{K}^+\hat{\mathbf{U}}, \quad \mathbf{B} = \mathbf{K}^T\hat{\mathbf{U}},$$

where $\hat{\mathbf{U}} \in \mathbb{R}^{T \times R}$ is the matrix having the top R left singular vectors of $\mathbf{K}\mathbf{Q}^\top$ as columns.

Proof. Observe that $\mathbf{K}\mathbf{A}\mathbf{B}^\top\mathbf{Q}^\top$ has rank at most R. Hence,

$$\min_{\mathbf{A},\mathbf{B} \in \mathbb{R}^{d \times R}} \|\mathbf{K}\mathbf{A}\mathbf{B}^\top\mathbf{Q}^\top - \mathbf{K}\mathbf{Q}^\top\|_F,$$

is lower bounded by

$$\min_{\mathbf{M} \in \mathbb{R}^{T \times T}} \|\mathbf{M} - \mathbf{K}\mathbf{Q}^\top\|_F \text{ s.t. } \operatorname{rank}(\mathbf{M}) \leq R.$$

By the Eckart-Young theorem, we know the best rank R approximation of $\mathbf{K}\mathbf{Q}^\top$ is given by its truncated SVD:

$$\mathbf{K}\mathbf{Q}^\top \simeq \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^\top,$$

where $\hat{\mathbf{U}} \in \mathbb{R}^{T \times R}, \hat{\mathbf{\Sigma}} \in \mathbb{R}^{R \times R}$ and $\hat{\mathbf{V}} \in \mathbb{R}^{T \times R}$. We will show that $\mathbf{K}\mathbf{A}\mathbf{B}^\top\mathbf{Q}^\top = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^\top$ from which the optimality of $\mathbf{A}$ and $\mathbf{B}$ follows.

Let $\mathbf{K} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$ be the full SVD of $\mathbf{K}$. Observe that we have the inclusion of (column) spans: $\mathcal{R}(\hat{\mathbf{U}}) \subseteq \mathcal{R}(\mathbf{U}) = \mathcal{R}(\mathbf{K})$. Since $\mathbf{K}\mathbf{K}^+$ is the orthogonal projection onto $\mathcal{R}(\mathbf{K})$, we have $\mathbf{K}\mathbf{K}^+\hat{\mathbf{U}} = \hat{\mathbf{U}}$, hence

$$\begin{aligned}\mathbf{K}\mathbf{A}\mathbf{B}^\top\mathbf{Q}^\top &= \mathbf{K}\mathbf{K}^+\hat{\mathbf{U}}\hat{\mathbf{U}}^\top\mathbf{K}\mathbf{Q}^\top = \hat{\mathbf{U}}\hat{\mathbf{U}}^\top\mathbf{K}\mathbf{Q}^\top \\ &= \hat{\mathbf{U}}\hat{\mathbf{U}}^\top\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^\top.\end{aligned}$$

Therefore $\mathbf{A} = \mathbf{K}^+\hat{\mathbf{U}}$ and $\mathbf{B} = \mathbf{K}^\top\hat{\mathbf{U}}$ are optimal solutions to $\min_{\mathbf{A},\mathbf{B} \in \mathbb{R}^{d \times R}} \|\mathbf{K}\mathbf{A}\mathbf{B}^\top\mathbf{Q}^\top - \mathbf{K}\mathbf{Q}^\top\|_F$. □

The Moore-Penrose pseudo inverse of $\mathbf{K}$ can be expressed through the SVD of $\mathbf{K}$ as $\mathbf{K}^+ = \mathbf{V}_K\mathbf{\Sigma}_K^{-1}\mathbf{U}_K^\top$. The singular value decomposition of $\mathbf{K}\mathbf{Q}^\top \in \mathbb{R}^{T \times T}$ can be computed efficiently as its rank is at most d. Indeed, we can first perform an SVD on $\mathbf{K} = \mathbf{U}_K\mathbf{\Sigma}_K\mathbf{V}_K^\top$ and $\mathbf{Q} = \mathbf{U}_Q\mathbf{\Sigma}_Q\mathbf{V}_Q^\top$, then on the $d \times d$ matrix $\mathbf{\Sigma}_K\mathbf{V}_K^\top\mathbf{V}_Q\mathbf{\Sigma}_Q = \mathbf{U}'\mathbf{\Sigma}'\mathbf{V}'^\top$. The SVD of $\mathbf{K}\mathbf{Q}^\top$ is then $\mathbf{U}_K\mathbf{U}'\mathbf{\Sigma}'(\mathbf{V}_Q\mathbf{V}')^\top = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ with $\mathbf{U} = \mathbf{U}_K\mathbf{U}' \in \mathbb{R}^{T \times d}$ and $\mathbf{V} = \mathbf{V}_Q\mathbf{V}' \in \mathbb{R}^{T \times d}$. This way the optimal solution provided by Theorem 2 can be computed efficiently in time $\mathcal{O}(Td^2)$.

A similar approach is used by KQ-SVD to derive optimal projections for the low rank approximation of the product of the value cache with the output matrix (see Appendix B).

5 THEORETICAL ANALYSIS

In this section, we focus on the minimization problem in Eq. (5); the same reasoning applies to value and output matrices (see Appendix B). We first provide an exact formula quantifying the accuracy difference between K-SVD and KQ-SVD. We then exhibit a failure mode of Eigen method [Saxena et al., 2024] which KQ-SVD circumvent by design. Finally, we show how KQ-SVD also obtains optimal low rank approximation in the Grouped Query Attention (GQA) framework.

5.1 Comparing K-SVD and KQ-SVD

In this section, we characterize the optimality gap between KQ-SVD and K-SVD (Section 3.3) and derive a closed form expression of their accuracy difference.

Theorem 3. *Let* $\text{opt} = \min_{\mathbf{A},\mathbf{B}\in\mathbb{R}^{d\times R}} \|\mathbf{K}\mathbf{A}\mathbf{B}^\top\mathbf{Q}^\top - \mathbf{K}\mathbf{Q}^\top\|_F^2$ *be the optimal error for the low rank attention approximation problem (achieved by KQ-SVD), and let* $\text{err}_{K\text{-}SVD} = \|\mathbf{K}\hat{\mathbf{V}}_K\hat{\mathbf{V}}_K^T\mathbf{Q}^T - \mathbf{K}\mathbf{Q}^T\|_F^2$ *be the error of K-SVD. We have*

$$\text{err}_{K\text{-}SVD} - \text{opt} = \sum_{i=1}^{R}\sigma_i(\mathbf{K}\mathbf{Q}^\top)^2 - \|\mathbf{K}\hat{\mathbf{V}}_K\hat{\mathbf{V}}_K^T\mathbf{Q}^T\|_F^2 \geq 0,$$

with equality only if the top R left singular vectors of $\mathbf{K}$ *and the top R of* $\mathbf{K}\mathbf{Q}^\top$ *span the same subspace.*

Proof. Let $\mathbf{K} \approx \hat{\mathbf{U}}_K\hat{\mathbf{\Sigma}}_K\hat{\mathbf{V}}_K^\top$ be the rank R truncated SVD of $\mathbf{K}$ and $\mathbf{K}\mathbf{Q}^\top \approx \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^\top$ be the one of $\mathbf{K}\mathbf{Q}^\top$.

By Theorem 2, $\text{opt} = \|\hat{\mathbf{U}}\hat{\mathbf{U}}^\top\mathbf{K}\mathbf{Q}^T - \mathbf{K}\mathbf{Q}^T\|_F^2 = \sum_{i=R+1}^{d}\sigma_i(\mathbf{K}\mathbf{Q}^T)^2$.

On the one hand we have

$$\|\mathbf{K}\mathbf{Q}^T\|_F^2 = \sum_{i=1}^{d}\sigma_i(\mathbf{K}\mathbf{Q}^T)^2 = \sum_{i=1}^{R}\sigma_i(\mathbf{K}\mathbf{Q}^T)^2 + \text{opt}.$$

On the other hand,

$$\|\mathbf{K}\mathbf{Q}^\top\|_F^2 = \|\mathbf{K}\mathbf{Q}^\top - \hat{\mathbf{U}}_K\hat{\mathbf{U}}_K^\top\mathbf{K}\mathbf{Q}^\top + \hat{\mathbf{U}}_K\hat{\mathbf{U}}_K^\top\mathbf{K}\mathbf{Q}^\top\|_F^2$$
$$= \|(\mathbf{I} - \hat{\mathbf{U}}_K\hat{\mathbf{U}}_K^\top)\mathbf{K}\mathbf{Q}^\top\|_F^2 + \|\hat{\mathbf{U}}_K\hat{\mathbf{U}}_K^\top\mathbf{K}\mathbf{Q}^\top\|_F^2,$$

since $(\mathbf{I} - \hat{\mathbf{U}}_K\hat{\mathbf{U}}_K^\top)$ and $\hat{\mathbf{U}}_K\hat{\mathbf{U}}_K^\top$ are projections on orthogonal subspaces.

Since $\mathbf{K}\hat{\mathbf{V}}_K\hat{\mathbf{V}}_K^\top = \hat{\mathbf{U}}_K\hat{\mathbf{U}}_K^\top\mathbf{K}$, the left term is exactly $\text{err}_{\text{K-SVD}}$ and the second term can be reduced to $\hat{\mathbf{U}}_K\hat{\mathbf{U}}_K^\top\mathbf{K}\mathbf{Q}^\top = \mathbf{K}\hat{\mathbf{V}}_K\hat{\mathbf{V}}_K^\top\mathbf{Q}^\top$.

Putting everything together, we get

$$\sum_{i=1}^{R}\sigma_i(\mathbf{K}\mathbf{Q}^T)^2 + \text{opt} = \text{err}_{\text{K-SVD}} + \|\mathbf{K}\hat{\mathbf{V}}_K\hat{\mathbf{V}}_K^\top\mathbf{Q}^\top\|_F^2,$$

which shows the equality in the theorem.

To show that $\text{err}_{\text{K-SVD}} - \text{opt} \geq 0$, first observe that

$$\text{err}_{\text{K-SVD}} - \text{opt} = \|\hat{\mathbf{U}}\hat{\mathbf{U}}^\top\mathbf{K}\mathbf{Q}^\top\| - \|\hat{\mathbf{U}}_K\hat{\mathbf{U}}_K^\top\mathbf{K}\mathbf{Q}^\top\|.$$

A direct consequence of the Eckart-Young theorem is that, for any matrix $\mathbf{M} \in \mathbb{R}^{T\times T}$ and any $R \leq T$, the solution of

$$\max_{\mathbf{X}\in\mathbb{R}^{T\times R}} \|\mathbf{X}\mathbf{X}^\top\mathbf{M}\| \text{ s.t. } \mathbf{X}^\top\mathbf{X} = \mathbf{I}$$

is obtained by setting the columns of $\mathbf{X}$ to the top R left singular vectors of $\mathbf{M}$. Hence, $\|\hat{\mathbf{U}}\hat{\mathbf{U}}^\top\mathbf{K}\mathbf{Q}^\top\| \geq \|\hat{\mathbf{U}}_K\hat{\mathbf{U}}_K^\top\mathbf{K}\mathbf{Q}^\top\|$ and thus $\text{err}_{\text{K-SVD}} - \text{opt} \geq 0$, with equality only if $\hat{\mathbf{U}}\hat{\mathbf{U}}^\top = \hat{\mathbf{U}}_K\hat{\mathbf{U}}_K^\top$, i.e., when the top R left singular vectors of $\mathbf{K}$ and the top R of $\mathbf{K}\mathbf{Q}^\top$ span the same subspace. □

It is worth observing that equality between $\text{err}_{\text{K-SVD}}$ and opt happens *only when* the projection onto $\hat{\mathbf{V}}_K$ captures *all* of the energy (Frobenius norm) in the top R singular values of $\mathbf{K}\mathbf{Q}^\top$. Since the best rank-$R$ approximation in the Frobenius norm is unique, this holds precisely when the subspace spanned by the top R left singular vectors of $\mathbf{K}$ coincides with that spanned by the top R left singular vectors of $\mathbf{K}\mathbf{Q}^\top$. In other words, equality holds *precisely when these two subspaces match.*

5.2 Comparing Eigen and KQ-SVD

We now compare Eigen [Saxena et al., 2024] with KQ-SVD. Although we do not derive an exact value for the optimality gap of Eigen, we identify a critical limitation of Eigen: the method is highly sensitive to unbalance between the norms of $\mathbf{K}$ and $\mathbf{Q}$. Eigen's performance can be degraded simply by multiplying $\mathbf{K}$ by a constant β and dividing $\mathbf{Q}$ by the same constant. While this rescaling leaves the attention computation unchanged and does not affect KQ-SVD, it causes Eigen method to behave increasingly like K-SVD as the unbalance between the two norms grows. Theorem 4 formalizes this intuition.

Theorem 4. *Let* $\mathbf{K}, \mathbf{Q} \in \mathbb{R}^{T\times d}$*, let* $\alpha = \frac{\|\mathbf{Q}\|_F}{\|\mathbf{K}\|_F}$ *and let* $\text{err}_{Eigen} = \|\mathbf{K}\hat{\mathbf{V}}_{Eigen}\hat{\mathbf{V}}_{Eigen}^\top\mathbf{Q}^\top - \mathbf{K}\mathbf{Q}^\top\|$ *be the error of Eigen. If there is a non-trivial gap between the Rth and $(R+1)$th singular values of* $\mathbf{K}$*, i.e.* $\sigma_R(\mathbf{K}) > \sigma_{R+1}(\mathbf{K})$*, then* $\lim_{\alpha\to 0}\text{err}_{Eigen} = \text{err}_{K\text{-}SVD}$.

Proof. Recall that Eigen approximate $\mathbf{K}\mathbf{Q}^\top$ with $\mathbf{K}\hat{\mathbf{V}}_{Eig}\hat{\mathbf{V}}_{Eig}^\top\mathbf{Q}^\top$ where $\hat{\mathbf{V}}_{Eig}$ is the matrix with the top R right singular vectors of $\begin{bmatrix}\mathbf{K}\\ \mathbf{Q}\end{bmatrix}$ as columns. In the limit where $\frac{\|\mathbf{Q}\|_F}{\|\mathbf{K}\|_F}$ tends to 0, the concatenated matrix $\begin{bmatrix}\mathbf{K}\\ \mathbf{Q}\end{bmatrix}$ tends to $\begin{bmatrix}\mathbf{K}\\ \mathbf{0}\end{bmatrix}$. Since $\sigma_R(\mathbf{K}) > \sigma_{R+1}(\mathbf{K})$, it follows from the Davis-Kahan theorem [Davis and Kahan, 1970] that, as α tends to 0, the space spanned by the top R right singular vectors of $\begin{bmatrix}\mathbf{K}\\ \mathbf{Q}\end{bmatrix}$ converges to the one of $\begin{bmatrix}\mathbf{K}\\ \mathbf{0}\end{bmatrix}$, and thus of $\mathbf{K}$. Hence $\lim_{\alpha\to 0}\hat{\mathbf{V}}_{Eig}\hat{\mathbf{V}}_{Eig}^\top = \hat{\mathbf{V}}_K\hat{\mathbf{V}}_K^\top$ and $\mathrm{err}_{\text{Eigen}}$ tends to $\mathrm{err}_{\text{K-SVD}}$. □

5.3 Handling grouped query attention

Standard multi-head attention (MHA) is powerful but slow at inference, while multi-query attention (MQA) [Shazeer, 2019] is much faster but can hurt model quality and requires retraining. Grouped-query attention (GQA) [Ainslie et al., 2023], sits in between MHA and MQA by letting groups of query heads share a key-value head, balancing accuracy and performance. GQA organizes attention/query heads into groups of size m. All query heads within a group share the same set of keys and values, enabling more efficient computation without significantly compromising model performance. We have assumed in the presentation of our method that each key head attends to a single query head. We show in the following theorem that simply applying KQ-SVD on the shared key cache and the concatenated query caches provides the optimal approximation with GQA.

Theorem 5. *Given a (shared) key cache matrix $\mathbf{K} \in \mathbb{R}^{T\times d}$ and m full column rank query cache matrices $\mathbf{Q}_1, \cdots, \mathbf{Q}_m \in \mathbb{R}^{T\times d}$, solving*

$$\min_{\mathbf{A},\mathbf{B}_1,\cdots,\mathbf{B}_m\in\mathbb{R}^{d\times R}} \sum_{i=1}^{m} \|\mathbf{K}\mathbf{A}\mathbf{B}_i^\top\mathbf{Q}_i^\top - \mathbf{K}\mathbf{Q}_i^\top\|_F^2,$$

is equivalent to solving

$$\min_{\mathbf{A},\mathbf{B}\in\mathbb{R}^{d\times R}} \|\mathbf{K}\mathbf{A}\mathbf{B}^\top\mathbf{Q}^\top - \mathbf{K}\mathbf{Q}^\top\|_F^2,$$

where $\mathbf{Q} = [\mathbf{Q}_1^\top\ \mathbf{Q}_2^\top\ \cdots\ \mathbf{Q}_m^\top]^\top \in \mathbb{R}^{mT\times d}$ is the matrix obtained by stacking the m query matrices.

Proof. We first show that the solution matrices $\mathbf{B}_i$ can be chosen to be all equal, i.e., that solving

$$\min_{\mathbf{A},\mathbf{B}_1,\cdots,\mathbf{B}_m\in\mathbb{R}^{d\times R}} \sum_{i=1}^{m} \|\mathbf{K}\mathbf{A}\mathbf{B}_i^\top\mathbf{Q}_i^\top - \mathbf{K}\mathbf{Q}_i^\top\|_F^2,$$

is equivalent to solving

$$\min_{\mathbf{A},\mathbf{B}\in\mathbb{R}^{d\times R}} \sum_{i=1}^{m} \|\mathbf{K}\mathbf{A}\mathbf{B}^\top\mathbf{Q}_i^\top - \mathbf{K}\mathbf{Q}_i^\top\|_F^2.$$

Indeed, for any $\mathbf{A}$ and any i, since $\mathbf{Q}_i^\top$ is full row rank, the minimizers of $\min_{\mathbf{B}_i}\|\mathbf{K}\mathbf{A}\mathbf{B}_i^\top\mathbf{Q}_i^\top - \mathbf{K}\mathbf{Q}_i^\top\|_F^2$ are the same as the minimizers of $\min_{\mathbf{B}_i}\|\mathbf{K}\mathbf{A}\mathbf{B}_i^\top - \mathbf{K}\|_F^2$, and are thus independent of $\mathbf{Q}_i$. Since this is true for any $\mathbf{A}$, and in particular for the optimal one, this shows that all the solutions $\mathbf{B}_i$ can be taken to be equal.

The result then directly follows from the fact that the squared Frobenius norm of a block matrix is equal to the sum of the squared Frobenius norms of the blocks. □

Theorem 5 states that optimal projections can be computed for models using GQA by stacking query matrices in each group and using KQ-SVD as in the non GQA case. Computing the optimal projection for a head group costs $\mathcal{O}(mTd^2)$, where m is the size of the head group, leading to an amortized cost per query head of $\mathcal{O}(Td^2)$.

6 EXPERIMENTS

The theoretical results established in the previous sections require further validation on real cache matrices generated by state-of-the-art LLMs. In this section, we first compare KQ-SVD with K-SVD and Eigen, and then demonstrate the claim of Theorem 4 using real-world cache data.

6.1 Comparing methods

Setup: To validate our theoretical claims, we run experiments on several widely used LLMs. We test two models without grouped-query attention (GQA)—Llama2-7B and Llama2-13B [Touvron et al., 2023]—and two models with GQA—Llama3-8B [Grattafiori et al., 2024] and Mistral-7B-v0.3 [Jiang et al., 2023]. All experiments use the C4 dataset [Raffel et al., 2020], with projections learned on the training split and evaluated on the validation split.

Learning projections: Following the methodology of [Saxena et al., 2024], we select 128 training sequences of 2048 tokens each from C4. Each sequence is passed through the model, storing the key, value, and query caches for every layer and attention head (queries are needed by both Eigen and KQ-SVD).

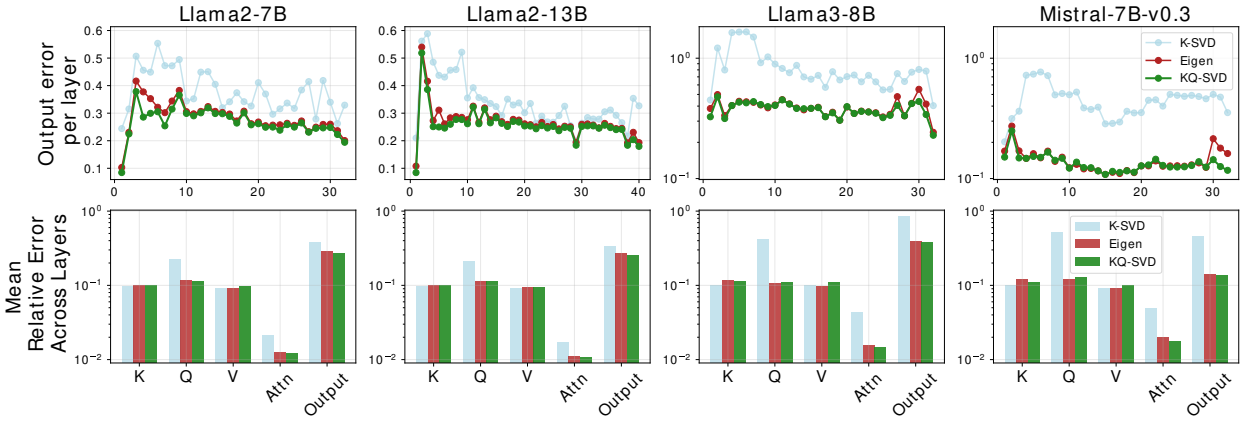

Figure 1: Relative Frobenius approximation output error per layer (top) and mean relative errors on $\mathbf{K}$, $\mathbf{Q}$, $\mathbf{V}$, $\mathbf{KQ}^\top$ and attention layer output across layers (bottom) for Mistral and Llama models.

For each layer-head pair, we collect the corresponding caches from all 128 sequences and concatenate them, yielding large matrices $\mathbf{K}, \mathbf{Q}, \mathbf{V} \in \mathbb{R}^{T_{\text{huge}} \times d}$ with $T_{\text{huge}} = 262{,}144$. Because model context length is limited and attention cost scales quadratically with sequence length, it is not feasible to build these matrices from a single long sequence. Instead, we construct them from multiple shorter ones. Once the large cache matrices are formed, we perform SVD and apply the formulas presented in previous sections to compute low-rank projections.

Rank selection: All methods are evaluated at the same rank R, determined individually for each layer. For a given layer, we analyze the singular value spectra of the key and value matrices, averaged across heads, and choose the smallest R that discards no more than an $\epsilon = 0.1$ fraction of the spectral energy. That is, for singular values $\{\sigma_j\}_j$ of $\mathbf{K}$, $\frac{\sum_{j=1}^{R} \sigma_j^2}{\sum_{i=1}^{d} \sigma_j^2} \geq 1 - \epsilon$. This is equivalent to requiring that the relative Frobenius error is at most ϵ.

Evaluation: We evaluate the learned projections on 32 validation sequences of 2048 tokens each. For every sequence, we extract the cache matrices $(\mathbf{K}, \mathbf{Q}, \mathbf{V})$ at each layer and head. Using these matrices, we can simulate attention computations directly, since attention depends only on these three components. Each cache is projected onto its corresponding low-rank subspace to form approximations, and we then compute the approximate Multi-Head Attention output using the standard formulas. For comparison, we also compute the exact (uncompressed) attention output. This enables us to measure the relative error of each method across all matrices of the attention pipeline. Errors are averaged across validation sequences.

Metrics: For a matrix $\mathbf{M}$ and its approximation $\widetilde{\mathbf{M}}$, we report the relative Frobenius norm error $\text{err}_{\text{Fro}} = \frac{\|\mathbf{M} - \widetilde{\mathbf{M}}\|_F^2}{\|\mathbf{M}\|_F^2}$. This error is computed for the key, query, and value matrices, for the attention score matrix $\mathbf{KQ}^\top$, and for the Multi-Head Attention output $\text{MHA}(\mathbf{X})$.

Results: Results are shown in Figure 1. For each model, the top plot reports the relative error on the attention output across layers, while the bottom plot reports the averaged errors on the intermediate components. We observe that K-SVD provides the most accurate approximation of the key matrices (as expected from the optimality of SVD), but performs poorly on query matrices, leading to weaker approximations of the attention scores and consequently higher output errors. This effect is more pronounced in GQA models, where sharing the key matrix across queries in a group amplifies approximation errors.

In contrast, Eigen and KQ-SVD achieve comparable accuracy on keys, queries, and values. The key difference lies in the attention score matrix: KQ-SVD consistently delivers higher accuracy, resulting in lower attention errors and outperforming all other methods on all models.

Performance on downstream tasks: We report results only in terms of the Frobenius-norm error of the attention matrices and outputs at each layer. Our experiments show that for an equivalent KV-cache compression ratio, our method consistently yields higher-fidelity attention outputs than competitors, and for a fixed error budget, our approach can achieve greater compression. Our underlying

assumption is that the error in the attention output directly correlates with downstream task performance. Papers such as Eigen [Saxena et al., 2024] provide evidence of this, showing for instance that on Llama-2-7B, perplexity increases from 5.47 to 7.48 under a 0.6x compression ratio, while accuracy decreases from 0.65 to 0.56. These results support the use of fidelity-based metrics as a reliable proxy for downstream performance, and suggest that our gains in reconstruction error would translate to consistent improvements on downstream tasks.

6.2 Unbalanced K and Q matrices

We verify experimentally the claims of Section 5.2 and analyze the attention approximation error under unbalanced $\mathbf{K}$ and $\mathbf{Q}$ matrices.

Set up: We follow the same experimental setup as in the previous section using the C4 train/validation split to learn projections and evaluate their accuracy. The only difference is that cache matrices are scaled to assess the effect of unbalance. Key matrices are multiplied by β and query matrices are divided by β. This is equivalent to scaling the projection matrices $\mathbf{W}_{K,i}$/ $\mathbf{W}_{Q,i}$ by β, as the operation permutes with matrix multiplication (this does not change the output since the query and key matrices are multiplied before any non-linear activation).

Metrics: For each unbalance ratio, we plot the relative attention output error for the three methods, averaged across all layers and validation sequences.

Results: Results are shown in Figure 2. As discussed in Section 5.2, K-SVD and KQ-SVD are invariant to scaling $\mathbf{K}$ by a factor β and dividing $\mathbf{Q}$ by the same factor, which is confirmed by the constant error observed for these methods. As predicted theoretically, a higher unbalance ratio brings Eigen closer to K-SVD; for $\beta = 10$, their errors are nearly indistinguishable. This confirms Theorem 4 and exposes a limitation of Eigen, which underperforms even under modest query-key unbalance.

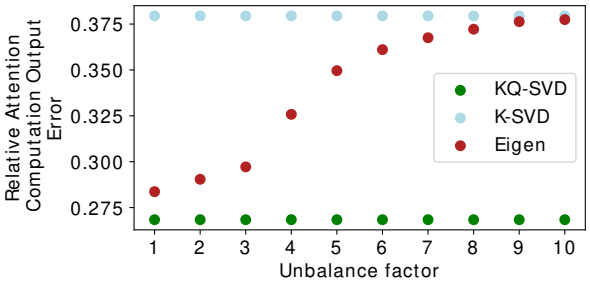

Figure 2: Llama2-7B relative output approximation error averaged across layers for varying unbalanced factor β.

7 CONCLUSION

We introduced KQ-SVD, a novel approach for computing low-rank projections for KV cache compression. KQ-SVD is driven by minimizing an upper bound on the attention output approximation error, resulting in an optimization problem that yields the optimal low-rank approximation of the attention matrix. Crucially, this problem admits a closed-form solution that can be computed efficiently. We quantify the advantage of KQ-SVD over previous methods, either through exact error formulas or by highlighting failure modes of prior approaches. Experiments validate our theoretical findings and demonstrate that considering the interaction between queries and keys—as KQ-SVD— provides a superior alternative to performing SVD on the key cache or on concatenated keys and queries. Our technique is complementary to popular cache compression methods such as GQA. An interesting following work could be to show that KQ-SVD is also compatible with other compression methods like quantization or token eviction.

Acknowledgment

This research is supported by the Canadian Institute for Advanced Research (CIFAR AI chair program). This work was completed while Damien Lesens interned at Mila. This work made use of compute resources provided by the Digital Research Alliance of Canada and by Mila (mila.quebec).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints, 2023. URL `https://arxiv.org/abs/2305.13245`.

Wael Alghamdi, Hsiang Hsu, Haewon Jeong, Hao Wang, P Winston Michalak, Shahab Asoodeh, and Flavio P Calmon. Beyond adult and compas:

Fairness in multi-class prediction. *arXiv preprint arXiv:2206.07801*, 2022.

Chi-Chih Chang, Chien-Yu Lin, Yash Akhauri, Wei-Cheng Lin, Kai-Chiang Wu, Luis Ceze, and Mohamed S Abdelfattah. xkv: Cross-layer svd for kv-cache compression. *arXiv preprint arXiv:2503.18893*, 2025a.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed S Abdelfattah, and Kai-Chiang Wu. Palu: Kv-cache compression with low-rank projection. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Devendra Singh Chaplot. Albert q. jiang, alexandre sablayrolles, arthur mensch, chris bamford, devendra singh chaplot, diego de las casas, florian bressand, gianna lengyel, guillaume lample, lucile saulnier, lélio renard lavaud, marie-anne lachaux, pierre stock, teven le scao, thibaut lavril, thomas wang, timothée lacroix, william el sayed. *arXiv preprint arXiv:2310.06825*, 3, 2023.

Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Yao Fu. Challenges in deploying long-context transformers: A theoretical peak performance analysis. *arXiv preprint arXiv:2405.08944*, 2024.

Nathan Godey, Alessio Devoto, Yu Zhao, Simone Scardapane, Pasquale Minervini, Éric Villemonte de la Clergerie, and Benoît Sagot. Q-filters: Leveraging query-key geometry for efficient key-value cache compression. In *Sparsity in LLMs (SLLM): Deep Dive into Mixture of Experts, Quantization, Hardware, and Inference*.

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.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL `http://jmlr.org/papers/v21/20-074.html`.

Utkarsh Saxena, Gobinda Saha, Sakshi Choudhary, and Kaushik Roy. Eigen attention: Attention in low-rank space for kv cache compression. *arXiv preprint arXiv:2408.05646*, 2024.

Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

Luohe Shi, Hongyi Zhang, Yao Yao, Zuchao Li, and Hai Zhao. Keep the cost down: A review on methods to optimize llm's kv-cache consumption. *arXiv preprint arXiv:2407.18003*, 2024.

Prajwal Singhania, Siddharth Singh, Shwai He, Soheil Feizi, and Abhinav Bhatele. Loki: Low-rank keys for efficient sparse attention. *Advances in Neural Information Processing Systems*, 37: 16692–16723, 2024.

Hanshi Sun, Li-Wen Chang, Wenlei Bao, Size Zheng, Ningxin Zheng, Xin Liu, Harry Dong, Yuejie Chi,

and Beidi Chen. Shadowkv: Kv cache in shadows for high-throughput long-context llm inference. *arXiv preprint arXiv:2410.21465*, 2024.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Hao Wang, Ligong Han, Kai Xu, and Akash Srivastava. Squat: Subspace-orthogonal kv cache quantization. *arXiv preprint arXiv:2503.24358*, 2025.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020. URL `https://arxiv.org/abs/1910.03771`.

Hao Yu, Zelan Yang, Shen Li, Yong Li, and Jianxin Wu. Effectively compress kv heads for llm. *arXiv preprint arXiv:2406.07056*, 2024.

Zhihang Yuan, Yuzhang Shang, Yue Song, Qiang Wu, Yan Yan, and Guangyu Sun. Asvd: Activation-aware singular value decomposition for compressing large language models, 2024. URL `https://arxiv.org/abs/2312.05821`.

Biao Zhang, Ivan Titov, and Rico Sennrich. Sparse attention with linear units. *arXiv preprint arXiv:2104.07012*, 2021.

Rongzhi Zhang, Kuang Wang, Liyuan Liu, Shuohang Wang, Hao Cheng, Chao Zhang, and Yelong Shen. Lorc: Low-rank compression for llms kv cache with a progressive compression strategy. *arXiv preprint arXiv:2410.03111*, 2024.

Zeyu Zhang and Haiying Shen. Zack: Zero-overhead llm inference acceleration via dimensionality compression of the key-value cache. *arXiv preprint arXiv:2408.04107*, 2024.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model.
   Yes, the setting and algorithms are explained in Section 3 and Section 4
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm.
   Yes
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries.
   Yes, provided in the supplemental material.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results.
   Yes
   (b) Complete proofs of all theoretical results.
   Yes
   (c) Clear explanations of any assumptions.
   Yes

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL).
   Yes, the instruction needed to reproduce the experimental results are provided in Section 6. The code will be available in the supplemental material.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen).
   Yes, training details are explained in the second point of 6.1.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times).
   Yes, the metrics we used are presented in the fifth point of Section 6.1.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider).

Not Applicable, we do not present experimental results that depend on the computing infrastructure used.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets.
   Yes, the authors of the models and datasets we use are cited in Section 6.

   (b) The license information of the assets, if applicable.
   Not applicable, the models and datasets we use are open-source.

   (c) New assets either in the supplemental material or as a URL, if applicable.
   Yes, the code will be in the supplemental materials

   (d) Information about consent from data providers/curators.
   Not Applicable

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content.
   Not Applicable

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots.
   Not Applicable

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable.
   Not Applicable

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation.
   Not Applicable

KQ-SVD: Compressing the KV Cache with Provable Guarantees on Attention Fidelity

(Supplementary Material)

A PROOF OF THEOREM 1

Theorem. *Let $\mathbf{X} \in \mathbb{R}^{T\times d}$ be a sequence of token embeddings, $\mathbf{K}, \mathbf{Q}, \mathbf{V} \in \mathbb{R}^{T\times d}$ and*

$$\mathrm{MHA}(\mathbf{X}) = \left[\mathrm{Softmax}(\mathbf{Q}_i\mathbf{K}_i^\top/\sqrt{d})\mathbf{V}_i)\right]_i \mathbf{W}^O,$$

$$\widetilde{\mathrm{MHA}}(\mathbf{X}) = \left[\mathrm{Softmax}(\mathbf{Q}_i\widetilde{\mathbf{K}_i}^\top/\sqrt{d})\widetilde{\mathbf{V}}_i)\right]_i \mathbf{W}^O,$$

where $\widetilde{\mathrm{MHA}}(\mathbf{X}), \widetilde{\mathbf{K}}_i$ and $\widetilde{\mathbf{V}}_i$ represent the approximation of $\mathrm{MHA}(\mathbf{X}), \mathbf{K}_i$ and $\mathbf{V}_i$, respectively. The difference between the actual attention output and the one produced with approximate keys and values is upper bounded as

$$\begin{aligned}
&\|\widetilde{\mathrm{MHA}}(\mathbf{X}) - \mathrm{MHA}(\mathbf{X})\|_2 \\
&\leq \sum_{i=1}^{h} \frac{\|\mathbf{V}_i\mathbf{W}_i^O\|_2}{\sqrt{d}} \|\mathbf{Q}_i\mathbf{K}_i^\top - \mathbf{Q}_i\widetilde{\mathbf{K}_i}^\top\|_2 \\
&+ \|\mathbf{V}_i\mathbf{W}_i^O - \widetilde{\mathbf{V}}_i\mathbf{W}_i^O\|_2.
\end{aligned} \tag{3}$$

Proof. Let $\mathbf{W}^O = [\mathbf{W}_1^O; \mathbf{W}_2^O; \cdots; \mathbf{W}_h^O] \in \mathbb{R}^{D\times D}$, where $\mathbf{W}_i^O \in \mathbb{R}^{d\times D}$ are stacked vertically. By the definition of multi-head attention (see Section 3.2), we can write

$$\mathrm{MHA}(\mathbf{X}) = [\mathbf{H}_1, \ldots, \mathbf{H}_h]\mathbf{W}^O = \sum_{i=1}^{h} \mathbf{H}_i\mathbf{W}_i^O, \quad \widetilde{\mathrm{MHA}}(\mathbf{X}) = [\widetilde{\mathbf{H}}_1, \ldots, \widetilde{\mathbf{H}}_h]\mathbf{W}^O = \sum_{i=1}^{h} \widetilde{\mathbf{H}}_i\mathbf{W}_i^O.$$

Therefore,

$$\|\widetilde{\mathrm{MHA}}(\mathbf{X}) - \mathrm{MHA}(\mathbf{X})\|_2 = \left\|\sum_{i=1}^{h} (\widetilde{\mathbf{H}}_i - \mathbf{H}_i)\mathbf{W}_i^O\right\|_2 \leq \sum_{i=1}^{h} \|(\widetilde{\mathbf{H}}_i - \mathbf{H}_i)\mathbf{W}_i^O\|_2.$$

For each head $i \in \{1, \ldots, h\}$,

$$\begin{aligned}
\|(\widetilde{\mathbf{H}}_i - \mathbf{H}_i)\mathbf{W}_i^O\|_2 &= \left\|\left(\mathrm{Softmax}(\tfrac{\mathbf{Q}_i\mathbf{K}_i^\top}{\sqrt{d}})\mathbf{V}_i - \mathrm{Softmax}(\tfrac{\mathbf{Q}_i\widetilde{\mathbf{K}}_i^\top}{\sqrt{d}})\widetilde{\mathbf{V}}_i\right)\mathbf{W}_i^O\right\|_2 \\
&\leq \left\|\left(\mathrm{Softmax}(\tfrac{\mathbf{Q}_i\mathbf{K}_i^\top}{\sqrt{d}}) - \mathrm{Softmax}(\tfrac{\mathbf{Q}_i\widetilde{\mathbf{K}}_i^\top}{\sqrt{d}})\right)\mathbf{V}_i\mathbf{W}_i^O\right\|_2 + \left\|\mathrm{Softmax}(\tfrac{\mathbf{Q}_i\widetilde{\mathbf{K}}_i^\top}{\sqrt{d}})(\mathbf{V}_i - \widetilde{\mathbf{V}}_i)\mathbf{W}_i^O\right\|_2.
\end{aligned} \tag{4}$$

For the first term in (4), applying the submultiplicative property of the 2-norm gives

$$\left\|\left(\mathrm{Softmax}(\tfrac{\mathbf{Q}_i\mathbf{K}_i^\top}{\sqrt{d}}) - \mathrm{Softmax}(\tfrac{\mathbf{Q}_i\widetilde{\mathbf{K}}_i^\top}{\sqrt{d}})\right)\mathbf{V}_i\mathbf{W}_i^O\right\|_2 \leq \|\mathrm{Softmax}(\tfrac{\mathbf{Q}_i\mathbf{K}_i^\top}{\sqrt{d}}) - \mathrm{Softmax}(\tfrac{\mathbf{Q}_i\widetilde{\mathbf{K}}_i^\top}{\sqrt{d}})\|_2 \, \|\mathbf{V}_i\mathbf{W}_i^O\|_2.$$

The factor $\|\mathbf{V}_i\mathbf{W}_i^O\|_2$ acts as an amplification term, capturing how sensitivity in the value projections may magnify small perturbations in the attention weights-this term is typically not directly controllable in practice.

Since the Softmax function is $\frac{1}{2}$-Lipschitz continuous (see Appendix A.4 [Alghamdi et al., 2022]), we have

$$\|\mathrm{Softmax}(\tfrac{\mathbf{Q}_i\mathbf{K}_i^\top}{\sqrt{d}}) - \mathrm{Softmax}(\tfrac{\mathbf{Q}_i\widetilde{\mathbf{K}}_i^\top}{\sqrt{d}})\|_2 \leq \frac{1}{\sqrt{d}}\|\mathbf{Q}_i\mathbf{K}_i^\top - \mathbf{Q}_i\widetilde{\mathbf{K}}_i^\top\|_2.$$

For the second term in (4), note that $\|\text{Softmax}(\cdot)\|_1 = 1$, implying $\|\text{Softmax}(\cdot)\|_2 \leq 1$. Therefore,

$$\left\|\text{Softmax}(\tfrac{\mathbf{Q}_i\widetilde{\mathbf{K}}_i^\top}{\sqrt{d}})(\mathbf{V}_i - \widetilde{\mathbf{V}}_i)\mathbf{W}_i^O\right\|_2 \leq \|\mathbf{V}_i\mathbf{W}_i^O - \widetilde{\mathbf{V}}_i\mathbf{W}_i^O\|_2.$$

Combining the two bounds yields

$$\|(\widetilde{\mathbf{H}}_i - \mathbf{H}_i)\mathbf{W}_i^O\|_2 \leq \frac{\|\mathbf{V}_i\mathbf{W}_i^O\|_2}{\sqrt{d}}\|\mathbf{Q}_i\mathbf{K}_i^\top - \mathbf{Q}_i\widetilde{\mathbf{K}}_i^\top\|_2 + \|\mathbf{V}_i\mathbf{W}_i^O - \widetilde{\mathbf{V}}_i\mathbf{W}_i^O\|_2.$$

Summing over all heads $i = 1, \ldots, h$ gives the desired bound (3), completing the proof. □

B VALUE-OUTPUT PROJECTION

In this section, we examine the interaction between the value representations and the output projection. While the main analysis of this paper focuses on the relationship between keys and queries, the same reasoning naturally extends to values $\mathbf{V}_i \in \mathbb{R}^{T\times d}$ and the output matrix $\mathbf{W}_i^O \in \mathbb{R}^{d\times D}$. To further tighten the upper bound established in Theorem 1, we aim to minimize the second term in the summation, which leads to the following optimization problem:

$$\min_{\widetilde{\mathbf{V}}} \left\|\mathbf{V}\mathbf{W}^O - \widetilde{\mathbf{V}}\mathbf{W}^O\right\|_F^2 \text{ s.t. } \text{rank}(\widetilde{\mathbf{V}}) \leq R,$$

where for simplicity we drop the subscript. Likewise for the keys and queries case, the optimization problem can be restated using a projection matrix $\mathbf{S} \in \mathbb{R}^{d\times d}$ such that

$$\min_{\mathbf{S}} \|\mathbf{V}\mathbf{S}\mathbf{W}^O - \mathbf{V}\mathbf{W}^O\|_F^2 \text{ s.t. } \text{rank}(\mathbf{S}) \leq R,$$

where $\mathbf{S} \in \mathbb{R}^{d\times d}$ where we write the projection matrix $\mathbf{S}$ as a product of two matrices, i.e., $\mathbf{S} = \mathbf{A}\mathbf{B}^\top$ with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{d\times R}$. The minimization problem tackled by KQ-SVD for the value and output projections is

$$\min_{\mathbf{A},\mathbf{B}\in\mathbb{R}^{d\times R}} \|\mathbf{V}\mathbf{A}\mathbf{B}^\top\mathbf{W}^O - \mathbf{V}\mathbf{W}^O\|_F^2. \tag{5}$$

Theorem 2 applies, in a similar fashion to the case of values and outputs.

C PRACTICAL SETTINGS

Code: The code used for experiments presented in Section 6 is available at `https://github.com/DamienLesens/KQ-SVD`. We used Pytorch [Paszke et al., 2019] and the Hugging Face `transformers` library [Wolf et al., 2020]. For reproducibility, we fixed random seeds equals to 0.

Hardware: All experiments were conducted on NVIDIA Tesla V100-SXM2-32GB GPUs with CUDA acceleration. The primary compute nodes were Intel Xeon E5-2698 v4 @ 2.20GHz (503GB RAM).