Activation Quantization of Vision Encoders Needs Prefixing Registers

Seunghyeon Kim[1], Taesun Yeom[1], Jinho Kim[2], Wonpyo Park[3], Kyuyeun Kim[3], and Jaeho Lee[1]*

[1] POSTECH, South Korea
{seunghyeon.kim, tsyeom, jaeho.lee}@postech.ac.kr
[2] NYU Langone Health, USA
Jinho.Kim@nyulangone.org
[3] Google, USA
{wppark, kyuyeunk}@google.com
https://github.com/spbob0418/RegCache

Abstract. Large pretrained vision encoders are central to multimodal intelligence, powering applications from on-device vision processing to vision-language models. Since these applications often demand real-time processing of massive visual data, reducing the inference cost of vision encoders is critical. Quantization offers a practical path, but it remains challenging even at 8-bit precision due to so-called outliers. In this work, we propose *RegCache*, a training-free algorithm that mitigates outliers in large-scale pretrained vision encoders and serves as a plug-in module that can be applied on top of other quantization methods. RegCache introduces outlier-prone yet semantically meaningless prefix tokens into the vision encoder, which prevent other tokens from having outliers. Notably, we observe that outliers in vision encoders behave differently from those in language models, motivating two technical innovations: middle-layer prefixing and token deletion. Experimental results show that our method consistently improves quantized model performance across various vision encoders, particularly in extremely low-bit regimes (e.g., 4-bit).

Keywords: Vision encoder · Activation quantization · Registers

1 Introduction

Transformer-based vision encoders, such as CLIP and DINOv2, lie at the core of modern multimodal systems [34, 36]. Leveraging the scalability of vision transformers (ViTs) [12], these models are pretrained on massive datasets using large-scale computation, yielding highly informative and versatile visual representations. However, their large size poses challenges for practical deployment. Vision encoders are frequently deployed as standalone models on edge devices, where

* Corresponding author.

storage limits and inference overhead become major bottlenecks [13].[4] Vision encoders also serve as the visual backbones of many vision-language models (VLMs) [2, 27], where their computational cost remains substantial, especially for high-resolution images or video [23, 42]. In video pipelines, the vision encoder can account for roughly 45% of overall latency, and may even exceed the LLM prefilling cost at high resolutions [23, 42].

To address these challenges, post-training quantization (PTQ) offers a practical solution, reducing memory usage and computational cost without additional training [7]. However, despite recent progress, PTQ methods for ViT-based vision encoders still suffer from significant performance degradation under low-bit quantization [24, 43, 50]. Moreover, unlike autoregressive large language models (LLMs), which are often memory-bound on GPU servers, vision encoders are typically non-autoregressive and thus more likely to operate in compute-bound regimes.[5] Quantizing both activations and weights therefore allows high-precision matrix multiplications to be replaced with low-precision operations (e.g., INT8), hence reducing computational cost.

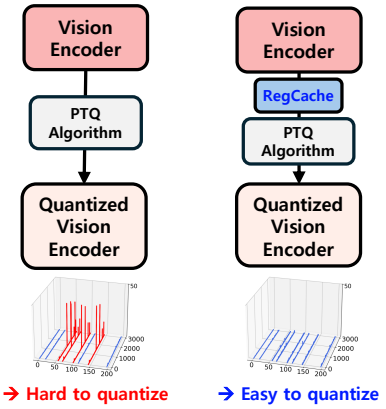

Fig. 1: A diagram of the proposed RegCache framework, pre-processing the vision encoder before PTQ.

However, activation quantization of large pretrained transformers is challenging due to *outlier* activations—i.e., a small number of extremely large activations that often arise in a few channels of later layers [39]. These outliers significantly expand the activation quantization range, leading to large quantization errors. While outlier-robust quantization has been widely studied for LLMs [11, 25, 44], existing methods typically assign different precision levels or quantization ranges to individual tokens or channels. This introduces substantial overhead, making them difficult to apply in static activation quantization settings [5, 38].

An emerging alternative is to directly mitigate outliers by prefixing *attention sink* tokens—i.e., semantically meaningless tokens such as ⟨BOS⟩ or ⟨SEP⟩ that absorb large attention from other tokens [39, 45]. Recent studies on LLM quantization show that inserting the activations of these sink tokens as a prefix in each attention layer can dramatically reduce the activation magnitudes of other tokens, thereby improving PTQ performance [5, 38, 47].

[4] To this end, several vendors provide deployment-ready lightweight vision encoders and toolchains for edge processors; see, e.g., NVIDIA (NV-CLIP), Qualcomm AI Hub (OpenAI CLIP), and Hailo (Hailo-CLIP).

[5] In contrast, the memory-bound nature of autoregressive LLM inference running on GPU servers renders weight-only quantization to be particularly effective.

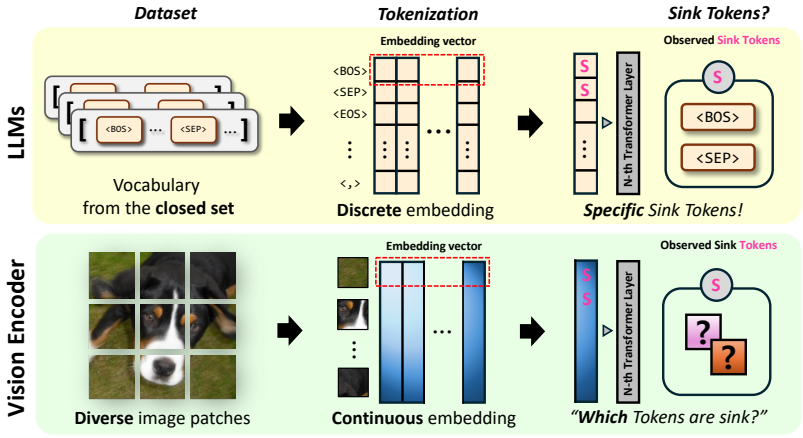

Fig. 2: Sink tokens in LLMs vs. vision encoders. Unlike LLMs, where sink tokens exist in a closed-set vocabulary, vision encoders process diverse image patches continuously mapped into an embedding space, making sink token discovery more challenging.

Naturally, one may ask: *Can we mitigate outliers in vision encoders by prefixing attention sinks*? Unfortunately, it remains unclear which token in a vision encoder (i.e., a patch embedding) could play a role analogous to attention sinks in language models. Unlike LLMs, typical vision encoders are not pretrained with tokens that are explicitly designed to be semantically meaningless (Fig. 2). Recent work suggests that introducing such meaningless tokens—termed *registers*—during training can improve the interpretability of ViT-based models [9]. However, incorporating registers remains uncommon in vision encoders.[6]

Contribution. In this work, we introduce ***RegCache*** (Register Caching), a novel prefix-based outlier mitigation algorithm for quantizing pretrained vision encoders. RegCache is inspired by the following empirical observation:

> Sink tokens emerge gradually from the **middle** layers of vision encoders, giving rise to outliers. These tokens are **highly similar across images**, enabling their use as **reusable registers** for other inputs at test time.

Based on these observations, which distinguish large pretrained vision encoders from relatively small ViTs, RegCache mitigates outliers by discovering and prefixing middle-layer registers in the form of a precomputed key-value (KV) cache. Notably, unlike in LLM prefixing [38], these tokens are inserted only in the middle-to-final layers and do not affect the early layers. RegCache also removes the remaining sink tokens that exhibit unusually large outliers. Through this process of adding and deleting tokens, RegCache replaces internally emerging sink

[6] In this regard, DINOv3 is a pleasant exception [37].

tokens with external precomputed caches, preventing them from inflating the activation quantization range. Importantly, this procedure requires no additional training, rendering RegCache a versatile go-to method. In practice, RegCache can be integrated as a lightweight on-top module into existing PTQ pipelines.

Throughout experiments, we apply RegCache to diverse text- and self-supervised vision encoders and combine it with recent PTQ methods for ViTs, which can be further improved by mitigating large activation outliers. We observe that such combinations consistently improve the prediction accuracy of quantized vision encoders across all considered setups. Notably, these gains are most pronounced under extremely low-bit quantization (e.g., 4-bit).

2 Related work

Outliers in large-scale transformers. In large-scale transformers, some activations in certain layers can attain magnitudes significantly larger than others—a phenomenon known as the *emergence of outliers* [3, 11, 21, 40]. Sun et al. [39] provide a systematic analysis and show that these outliers arise from the softmax operation in self-attention in both LLMs and ViTs. In LLMs, extreme activations are often concentrated on special tokens such as ⟨BOS⟩ or ⟨SEP⟩. In ViTs, several studies report that outlier tokens typically correspond to uninformative background patches, and removing them can improve internal representations [9, 19, 30]. However, it remains unclear which specific visual tokens consistently produce outliers in ViT-based vision encoders, as the corresponding patches vary across images. To address this gap, we observe that outlier tokens typically emerge in intermediate blocks and exhibit similar features across images and across a wide range of vision encoders, enabling them to be precomputed for downstream use cases such as PTQ.

Improving vision transformers via controlling attention sink tokens. Attention sinks, first highlighted by Xiao et al. [45], are tokens with little or no semantic content that nonetheless attract excessive attention in both LLMs [17] and ViTs [9]. In ViTs, these sink tokens act as noise in the attention map, hindering the model's ability to capture relations between patches and degrading downstream visual performance [9, 19, 20, 30]. To mitigate this issue, Darcet et al. [9] introduce an additional register token during training to absorb the attention sinks that emerge in image patches. More recently, Jiang et al. [19] propose a training-free *test-time register* that relocates the maximum activations from channels that frequently exhibit outliers into an extra token during inference. RegCache, in contrast, is specifically tailored to the quantization scheme. We establish a formal correlation between quantization sensitivity and outlier emergence, and exploit this relationship to suppress the outliers that degrade PTQ performance. Furthermore, our method relies on pre-computed token-level operations that incur minimal overhead and require no per-image processing, which is efficient and well-aligned with the goal of quantization.

Post-training quantization for vision transformers. Prior work has explored reducing the inference cost of large-scale ViT-based models through PTQ. Early methods mitigate quantization errors by assigning dynamic bitwidths to self-attention-sensitive blocks [29]. Subsequent studies attribute the low PTQ performance of ViTs to activation outliers arising from operations such as LayerNorm, softmax, and GELU. Motivated by this observation, RepQ-ViT [24] and PTQ4ViT [48] propose quantization schemes that isolate and minimize the impact of outliers. Other approaches address the heavy-tailed activation distribution; for example, NoisyQuant [28] injects noise to reshape the activation distribution into a more quantization-friendly form. FIMA-Q [43] introduces round-function optimization for PTQ, while ERQ [50] proposes a two-step procedure that sequentially quantizes activations and weights to reduce quantization error. Despite their methodological diversity, these approaches share a common premise: they are applied in the presence of outliers–whether through specialized quantizers for heavy-tailed activations [48, 50], scale reparameterization [24, 50], distribution reshaping [28], quantizer optimization [43], or ridge regression [50]. In contrast, RegCache directly suppresses outliers by leveraging a structural property of the vision encoder. As a result, RegCache can be easily integrated into existing PTQ pipelines to further reduce quantization error.

Outlier mitigation by prefixing sink-like tokens. Son et al. [38] mitigate outliers in LLMs by prefixing the sink-like tokens in the form of a KV cache. While RegCache inherits the high-level idea of prefixing register tokens, it introduces three technical innovations that account for the unique properties of ViTs. (1) RegCache constructs its register candidate set in a data-driven manner, whereas LLM methods simply use the dictionary, since visual tokens are continuous embeddings rather than entries of a fixed vocabulary. (2) While Son et al. [38] caches prefix tokens from the model input, where outliers arise in the early layers of LLMs, RegCache instead caches from the middle-layer, since ViT outliers emerge from the intermediate blocks (see 3.2). (3) RegCache additionally removes tokens to eliminate leftover outliers, which is critical to performance (see Tab. 7).

3 A closer look at outliers in vision encoders

Outliers in vision encoders tend to emerge at seemingly random background patch tokens, whereas in LLMs they typically occur at specific token positions or types [9, 39]. This lack of consistency makes it difficult to mitigate outliers in vision encoders through prefixing [38], as such strategies require caching tokens that consistently induce outliers across arbitrary test-time inputs (i.e., registers).

In this section, we present three key findings that motivate an alternative strategy—*identifying reusable sink tokens in the middle blocks of the model and prefixing them there*. These findings also shed light on why outliers in vision encoders predominantly emerge in the middle layers.

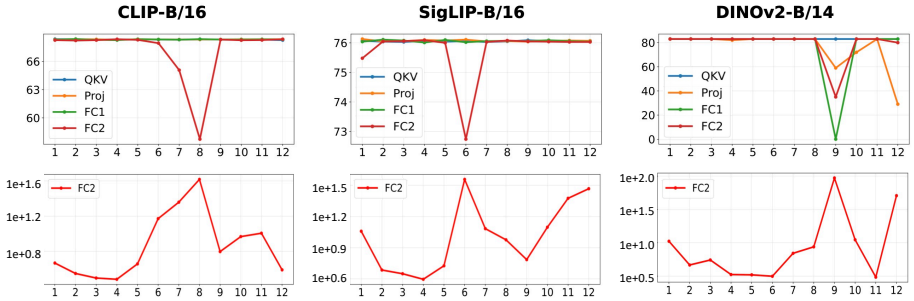

Fig. 3: (Top) Layerwise quantization sensitivity (%). We plot the zero-shot ImageNet-1k accuracy when quantizing each layer individually to W8A8. **(Bottom) Maximum norm of the FC2 layer input tokens for each layer.** We plot the largest ℓ_∞-norm across all tokens per image on a log scale, averaged over the ImageNet-1k validation set. For both plots, the x-axis denotes the transformer block index.

- Section 3.1: Quantization-sensitive layers primarily occur in the middle of the network, where outliers emerge, while remaining low elsewhere; thus, prefixing is unnecessary in early layers.
- Section 3.2: We explain why outliers emerge later in the network: early layers first identify semantically meaningless tokens.
- Section 3.3: In the middle layers, outlier tokens become highly similar across images, suggesting that they act as reusable registers.

3.1 Layerwise quantization sensitivity and outliers

We first analyze the quantization sensitivity of each layer in vision encoders and examine its connection to the emergence of activation outliers (i.e., FC2 inputs). In Fig. 3 (top), we report the layerwise quantization sensitivity, measured as the zero-shot ImageNet-1k accuracy when a given layer is quantized to W8A8. We observe that quantization-sensitive layers—i.e., layers that incur substantial accuracy drops when quantized—are localized to the MLP projection layers in one or two middle layers. In DINOv2, the performance degradation is particularly pronounced, and takes place in other layers as well. Moreover, as shown in Fig. 3 (bottom), these quantization-sensitive layers coincide with the blocks where activation outliers begin to emerge. Together, these suggest that outliers are a primary driver of performance degradation when quantizing vision encoders. In Appendix, we further analyze the effect of outlier tokens under quantization, highlighting the importance of mitigating them to improve PTQ performance.

In particular, we establish a clear connection between quantized accuracy and the high-norm activation behavior. Our findings further suggest that monitoring FC2 activations or directly measuring quantization sensitivity can help identify the blocks where prefixing should be applied. Additional results for other vision encoders (OpenCLIP and SigLIP2) are provided in Appendix.

3.2 Why the middle layers?

A natural question arises: Why do outliers in vision encoders emerge in the middle layers, whereas in LLMs they appear in the early layers? We hypothesize that this difference stems from the difficulty of identifying *semantically meaningless* tokens from raw image patches. Such tokens become distinguishable only after several processing blocks of the vision encoder. In contrast, in LLMs some tokens are clearly meaningless from the outset, such as ⟨BOS⟩, ⟨SEP⟩ (see Fig. 2, left).

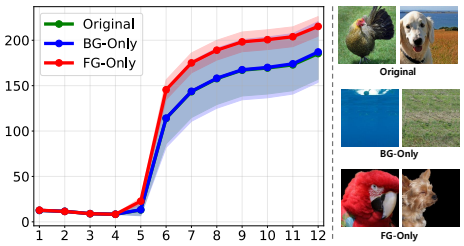

Fig. 4: Outliers in FG/BG-only images, for SigLIP-B/16 model. Note that "Original" and "BG-only" nearly overlap.

To test this hypothesis, we design an experiment comparing outlier emergence in images where meaningless patches are easily identifiable against those where the distinction is less clear. Specifically, using the test set of ImageNet-9 [46], we compare foreground-only images—where the background pixels are zero-ed out—with the originals. In foreground-only images, semantically meaningless patches should be easier to identify, requiring fewer processing blocks.

As shown in Fig. 4, outliers in foreground-only images indeed emerge earlier and with larger magnitude than in the original images, supporting our hypothesis. In contrast, removing the foreground and retaining only the background leaves the outlier behavior largely unchanged (i.e., the curves nearly overlap). In Appendix, we further analyze vision encoders trained with registers, where meaningless tokens are explicitly defined. These models exhibit behavior similar to LLMs, with outliers emerging from the early layers.

3.3 Reusability of outlier tokens

Next, we examine the outlier tokens in the quantization-sensitive layer. Specifically, we measure the cosine similarity between middle-layer FC2 input outlier tokens (i.e., those with the largest ℓ_∞-norm) extracted from pairs of images. Using 64 randomly sampled images from the ImageNet-1k validation split, we compute the mean pairwise cosine similarity.

As shown in Tab. 1, outlier tokens are highly similar across images, with a mean cosine similarity of 0.89. In contrast, normal tokens are much less similar, with an average of

Table 1: The average cosine similarity between two distinct groups of tokens in SigLIP-B/16.

Token Type	Cosine sim.
Normal tokens	0.26 (±0.10)
Outlier tokens	0.89 (±0.07)

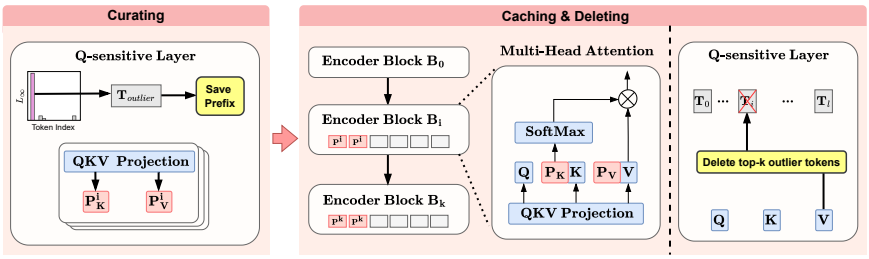

Fig. 5: An overview of the proposed RegCache algorithm. We identify a universal register by analyzing the inputs of quantization-sensitive layers across blocks. During inference, the register is inserted into each block, and outlier tokens are removed from the most quantization-sensitive layer.

only 0.26. This suggests that outliers contain components largely independent of the input image and may represent shared features that persist across samples. In Appendix, we provide a theoretical explanation, showing that it arises from the large magnitude of the outliers and the alignment of their locations.

4 Method

Before introducing our method, we recall two observations from Sec. 3:

- In vision encoders, outliers tend to emerge in the middle layers (e.g., fully-connected layers in blocks 5–8), whereas in LLMs they begin to appear in the early layers.
- Sink tokens (i.e., outlier-prone tokens) discovered in the middle layers are highly similar across images.

Combining these observations with prior findings in LLMs—prefixing additional sink tokens can mitigate outliers [38]—we arrive at the following hypothesis:

"Middle-layer sink tokens from one image can act as registers and help mitigate outliers in vision encoders processing another image."

Based on this hypothesis, we propose RegCache (Register Caching), an outlier-mitigation algorithm that replaces internally emerging sink tokens by inserting register tokens discovered from reference images. In essence, RegCache operates in three steps (see Fig. 5).

1. ***Curating*** a set of register candidate tokens—i.e., with large activation magnitudes—from a pool of reference images. (Sec. 4.1)
2. ***Caching*** the keys and values of selected register candidate tokens in the quantization-sensitive layers. (Sec. 4.2)
3. ***Deleting*** the sink tokens that have internally emerged—i.e., not inserted—to eliminate remaining outliers. (Sec. 4.3)

RegCache requires no additional training and performs only a few rounds of validation on a reference task and dataset, as will be discussed below. As a result, the algorithm requires neither large amounts of data nor substantial computational resources. At inference time, RegCache temporarily adds and removes a few tokens, introducing only negligible overhead (see Sec. 5.4).

4.1 Curating

Given a pretrained vision encoder, we first identify the quantization-sensitive layer of the model. We then construct a curated set of register candidate tokens by selecting the top-k tokens with the largest activation at the input of this layer from a pool of reference images.

Identifying the quantization-sensitive layer. Following Sec. 3.1, we quantize each layer of the vision encoder independently, and select the one that yields the largest accuracy drop on a reference task as the quantization-sensitive layer. If a specific base quantization algorithm is intended for deployment, we use it during this analysis; otherwise, we use standard round-to-nearest quantization. As the reference task, we use ImageNet-1k classification (on the training split), which is widely regarded as representative of visual understanding. As a label-free alternative, we also provide a reconstruction-loss-based approach that leverages unlabeled data in Appendix.

Curating the set of register candidates. After identifying the quantization-sensitive layer, we construct a set of register candidate tokens likely to act as registers when inserted near this layer. Specifically, we run inference on a pool of reference images and select the tokens with the largest ℓ_∞ norm at the quantization-sensitive layer. Formally, let l_{q} denote the quantization-sensitive layer identified via the sensitivity analysis in Sec. 3.1, and let $\Phi_l(\mathbf{x})$ denote the set of tokens at the input of the l-th layer for an image $\mathbf{x}$. We construct the register candidate set as

$$\mathcal{S} = \operatorname{argtop}k \left\{ \|\mathbf{z}\|_\infty \;\middle|\; \mathbf{z} \in \Phi_{l_q}(\mathbf{x}), \quad \text{for some } \mathbf{x} \in \mathcal{I}_{\mathrm{ref}} \right\}, \tag{1}$$

where $\mathcal{I}_{\mathrm{ref}}$ denotes the pool of reference images. In our experiments, we sample 50,000 images from the ImageNet-1k training split and set $k = 100$. Since sink tokens often emerge several blocks before the quantization-sensitive layer, we perform the same search for up to three preceding blocks, constructing separate register candidate sets for each block.

4.2 Caching

Having the set of register candidates $\mathcal{S}$ constructed, we compose the register by simply averaging the KV caches of the register candidate tokens $\mathbf{z}^* \in \mathcal{S}$. We then search for the optimal $\tau^* \in \mathbb{N}$, the number of times the register is copied and inserted into the target vision encoder. Precisely, the search is done as follows.

- First, we compute KV caches for each register candidate token $\mathbf{z} \in \mathcal{S}$, for blocks starting from the few layers before the quantization-sensitive block, as well as all subsequent blocks using the *unquantized* vision encoder.
- Then, we insert the averaged KV cache to the *quantized* vision encoder, with different numbers of copies. We vary τ within the range $\{1, 2, \ldots, 15\}$ and select τ^* as the one with the highest reference task accuracy.[7] Here, as in Sec. 4.1, we consider classification on the training split of ImageNet-1k dataset as our reference task.

4.3 Deleting

Finally, we apply a *token deletion* process to the input of the quantization-sensitive block (i.e., the block where l_q is located) in the vision encoder. At the inference phase, this process removes the sink tokens that emerge among the image patch tokens, thus removing any remaining outliers. Precisely, given some test image $\mathbf{x}_{\text{test}}$, we select the tokens with the top-$\tilde{k}$ ℓ_∞ norm, *i.e.*,

$$\mathcal{D} = \text{argtop}\tilde{\text{k}}\big\{ \|\mathbf{z}\|_\infty \;\big|\; \mathbf{z} \in \Phi_{l_q}(\mathbf{x}_{\text{test}}) \big\} \tag{2}$$

and remove these tokens. Here, similarly to the curating and caching steps, the number of tokens to be removed—*i.e.*, $\tilde{k}$—is tuned using the reference task.

5 Experiments

In this section, we first describe our experimental setup in Sec. 5.1. Next, we report results for both standalone vision encoders and VLMs across various benchmarks in Sec. 5.2, demonstrating consistent gains from RegCache over baselines including low-bit regimes. We also assess the generalizability of the precomputed prefix tokens and further validate our design through analyses of prefix-token behavior, ablation studies, and latency measurements.

5.1 Setup

Standalone vision encoders. We evaluate the proposed method on five widely used vision encoders: (1) CLIP [36], (2) OpenCLIP [6], (3) SigLIP [49], (4) SigLIP2 [41], and (5) DINOv2 [34], which cover diverse training configurations. For evaluation, we measure the quality of the quantized vision encoders using zero-shot accuracy on two downstream tasks: (1) Image classification on ImageNet-1k [10] and (2) text-image retrieval on MS-COCO [26]. To further validate the generalizability of the prefix searched from ImageNet-1k, we evaluate on diverse classification benchmarks, including Stanford Cars [22], Flowers-102 [31], Food-101 [4], CIFAR-100, Caltech101 [14], Oxford-IIIT Pet [35], and DTD [8].

[7] The total search cost of our algorithm is about 1 hour when using a naïve RTN pseudo-quantization algorithm on an RTX 4090.

Table 2: Zero-shot classification accuracy on ImageNet-1k. We have used various base quantization algorithms to quantize to 4/6/8 bits. The best results are marked in **bold**. Best/Average Δ denote the maximum/average accuracy gaps between each baseline and its RegCache counterpart, excluding the Naïve cases.

	CLIP-B/16			OpenCLIP-B/16			SigLIP-B/16			SigLIP2-B/16			DINOv2-B/14		
Method	**W4A4**	**W6A6**	**W8A8**	**W4A4**	**W6A6**	**W8A8**	**W4A4**	**W6A6**	**W8A8**	**W4A4**	**W6A6**	**W8A8**	**W4A4**	**W6A6**	**W8A8**
FP32		68.32			70.22			76.05			78.47			83.26	
Naïve	0.09	0.17	34.01	0.10	0.47	46.12	0.13	0.77	69.71	0.14	0.19	26.04	0.00	0.01	19.20
w/ RegCache	0.07	0.44	59.71	0.10	1.20	66.90	0.10	24.41	74.38	0.11	0.26	72.35	0.18	0.42	22.34
PTQ4ViT	0.37	51.60	67.69	0.09	59.98	69.39	0.19	68.68	75.57	0.28	41.54	76.92	0.01	78.48	**82.97**
w/ RegCache	1.78	55.49	67.86	0.14	63.60	69.60	0.59	72.38	75.75	2.67	69.02	76.88	2.38	**78.55**	82.92
RepQ-ViT	1.83	53.25	67.39	1.18	46.51	68.70	21.36	73.32	75.23	0.40	64.91	76.43	4.52	19.61	82.27
w/ RegCache	21.50	66.73	**68.10**	11.10	68.22	70.06	33.76	74.52	75.94	10.15	75.40	77.13	4.97	22.38	81.55
NoisyQuant	0.34	46.19	63.20	2.50	59.05	67.08	1.78	71.10	75.50	0.46	44.50	70.83	1.61	49.25	71.46
w/ RegCache	1.42	57.41	65.82	10.41	67.51	69.36	9.19	72.28	75.64	0.54	62.60	76.44	2.04	48.65	70.38
FIMA-Q	50.41	66.51	67.63	60.09	67.24	68.53	76.05	76.06	76.06	**78.48**	**78.47**	**78.48**	OOM	OOM	OOM
w/ RegCache	**62.08**	66.70	67.86	**65.11**	68.62	69.13	**76.13**	**76.12**	**76.12**	78.38	78.37	78.37	OOM	OOM	OOM
ERQ	1.56	39.64	67.99	43.15	66.70	69.25	57.76	74.44	75.95	10.56	74.75	78.05	6.53	78.03	82.59
w/ RegCache	46.07	**66.79**	68.09	54.76	**69.09**	**70.12**	60.99	75.25	75.98	28.98	75.32	78.45	**9.15**	77.88	82.54
Best Δ	+44.51	+27.15	+2.62	+11.61	+21.84	+2.28	+12.40	+3.38	+0.71	+18.42	+27.48	+5.61	+2.62	+2.77	-0.05
Average Δ	+15.67	+11.19	+0.77	+6.90	+7.53	+1.06	+4.70	+1.35	+0.25	+6.11	+11.31	+1.36	1.11	+0.52	-0.47

Vision-language models. We primarily use Qwen3-VL [1] (2B and 8B) as a strong, widely adopted open-source VLM that can handle both image and video. Furthermore, to demonstrate the broad applicability of our method, we also evaluate it on other VLMs, such as LLaVA [27] in Appendix.

We evaluate image understanding on GQA [18] and VQAv2 [16], as visual question-answering benchmarks. For video understanding, we use Video-MME [15] to assess multi-domain reasoning and MLVU [51], which includes long-video evaluation where vision-encoder overhead can become a critical bottleneck.

Table 3: Zero-shot image–text retrieval performance of CLIP and SigLIP on MS-COCO (R@1). The results are reported in the same format as in Tab. 2.

	CLIP-B/16						SigLIP-B/16					
	Image → Text			Text → Image			Image → Text			Text → Image		
	W4A4	W6A6	W8A8	W4A4	W6A6	W8A8	W4A4	W6A6	W8A8	W4A4	W6A6	W8A8
FP32	52.94	52.94	52.94	32.73	32.73	32.73	67.68	67.68	67.68	47.19	47.19	47.19
Naïve	0.00	0.00	22.76	0.01	0.06	14.08	0.00	0.34	60.04	0.04	0.86	41.80
w/ RegCache	0.02	0.22	46.10	0.04	0.44	28.02	0.04	12.06	65.76	0.04	12.70	46.30
PTQ4ViT	0.06	37.28	52.78	0.17	23.00	32.00	0.20	60.66	66.86	0.78	41.73	47.16
w/ RegCache	0.52	43.46	53.54	0.14	26.51	32.48	2.10	61.42	67.72	1.78	43.05	47.61
RepQ-ViT	0.32	29.06	44.52	0.47	15.90	23.01	5.14	37.50	65.90	6.21	26.41	46.33
w/ RegCache	4.32	38.68	45.58	3.37	19.70	23.68	14.92	61.06	66.12	9.37	43.63	46.42
NoisyQuant	0.12	25.26	48.94	0.19	18.02	31.07	0.29	52.52	67.10	1.18	33.36	46.76
w/ RegCache	0.37	33.86	49.10	0.71	22.28	30.40	2.53	63.28	67.24	3.81	43.25	**47.64**
FIMA-Q	40.92	52.90	53.58	27.71	32.33	32.76	67.66	67.70	67.62	47.23	47.22	47.21
w/ RegCache	**51.82**	**53.22**	**53.66**	**33.43**	**32.88**	**33.01**	**68.14**	**68.14**	**68.18**	**47.64**	**47.62**	47.62
ERQ	0.32	23.62	44.78	3.71	15.58	23.31	28.96	61.94	65.80	20.70	43.48	46.46
w/ RegCache	13.78	40.88	44.86	6.52	19.82	23.52	32.66	62.30	66.16	23.21	44.61	46.63
Best Δ	+13.46	+17.26	+1.06	+5.72	+4.26	+0.67	+9.78	+23.56	+0.86	+5.41	+17.22	+0.88
Average Δ	+5.81	+8.40	+0.43	+2.19	+3.27	+0.19	+3.55	+7.18	+0.43	+3.10	+5.99	+0.40

Table 4: VLM performance. We evaluate our method on image and video benchmarks using Qwen3-VL-{2B,8B} models under 4-bit quantization.

Method	Qwen3-VL-2B				Qwen3-VL-8B			
	Image		Video		Image		Video	
	GQA	VQAv2	Video-MME	MLVU	GQA	VQAv2	Video-MME	MLVU
Full-precision	58.59	77.05	73.44	68.34	59.04	77.86	78.78	78.19
Naïve	27.12	10.86	32.11	37.40	22.32	26.14	35.33	39.33
RepQ-ViT	38.77	32.03	44.56	41.17	37.84	46.29	50.44	44.89
w/ RegCache	42.11 (+3.34)	42.72 (+10.69)	49.22 (+4.66)	43.47 (+2.30)	44.00 (+6.16)	50.44 (+4.15)	52.89 (+2.45)	45.40 (+0.51)

Base quantization algorithms and details. To assess the broad applicability of RegCache as an effective on-top method, we evaluate it on four distinct categories of ViT PTQ baselines: (1) scaling factor optimization (PTQ4ViT [48], RepQ-ViT [24], and FIMA-Q [43]), (2) rounding function optimization (ERQ [50] and FIMA-Q [43]), (3) weight correction (ERQ [50]), and (4) activation distribution shaping (NoisyQuant [28]) to reduce quantization error.

Additionally, for CLIP and SigLIP models, prefixes are inserted from the searched layer to the final layer. DINOv2, trained in a self-supervised manner, exhibits different behavior compared to CLIP and SigLIP models; consequently, we find that inserting the prefix only at the searched layer yields better results.

Further details on the experimental setup can be found in Appendix.

5.2 Main results

Standalone vision encoders. In Tab. 2, we report the zero-shot image classification accuracy on ImageNet-1k dataset. We observe that the baselines combined with RegCache consistently achieve better accuracy in most settings. Specifically, baselines with RegCache outperform the base quantization methods in terms of both best accuracy gap (Best Δ) and average accuracy gap (Average Δ). Only one setup—DINOv2—exhibits a negligible accuracy drop.

For zero-shot image–text retrieval (Tab. 3), we observe that RegCache consistently improves performance across all setups. These results indicate RegCache can integrate well with other quantization methods across diverse tasks.

In particular, RegCache benefits low-precision quantization (i.e., 4-bit and 6-bit) most, where activation outliers affect quantization more severely. Conversely, the gain shrinks when headroom is limited, e.g., at high bitwidths with advanced base PTQ methods that already compensate for quantization errors. The gain is also larger on softmax-based models (e.g., CLIP) than on SigLIP, consistent with the common understanding that softmax is a key contributor to outliers.

Vision-language models. We further evaluate our quantized vision encoders within VLMs on image and video benchmarks (Tab. 4). All results are reported in the 4-bit setting on top of the PTQ baseline (RepQ-ViT), reflecting a highly constrained deployment setting. As shown in Tab. 4, RegCache consistently improves accuracy in VLMs, demonstrating robust gains across benchmarks.

Table 5: Zero-shot classification accuracy (%) on various datasets.

Model	Method	StanfordCars	Flowers-102	Food-101	CIFAR-100	Caltech101	OxfordIIIPet	DTD
CLIP-B/16	FP32	64.41	65.88	85.22	68.44	85.82	88.03	44.31
	Naïve	29.76	26.20	33.30	35.96	70.59	74.19	30.05
	w/ RegCache	49.96 (+20.20)	55.39 (+29.19)	74.68 (+41.38)	51.87 (+15.91)	83.22 (+12.63)	86.21 (+12.02)	42.82 (+12.77)
OpenCLIP-B/16	FP32	88.07	69.88	83.77	76.82	88.37	88.74	54.89
	Naïve	74.85	42.97	36.44	40.61	75.44	78.09	36.28
	w/ RegCache	85.85 (+11.00)	68.06 (+25.09)	80.80 (+44.36)	71.73 (+31.12)	86.88 (+11.44)	87.00 (+8.91)	51.70 (+15.42)
SigLIP-B/16	FP32	90.81	82.63	89.34	72.33	93.10	92.15	63.51
	Naïve	87.97	75.26	78.31	54.79	92.65	90.24	62.82
	w/ RegCache	89.73 (+1.76)	80.32 (+5.06)	88.17 (+9.86)	66.87 (+12.08)	92.65 (+0.00)	91.33 (+1.09)	63.62 (+0.80)
SigLIP2-B/16	FP32	92.74	83.38	90.65	77.10	93.19	92.31	61.33
	Naïve	35.12	26.38	30.55	20.92	67.48	58.22	37.23
	w/ RegCache	88.20 (+53.08)	76.50 (+50.12)	86.47 (+55.92)	59.78 (+38.86)	92.26 (+24.78)	89.70 (+31.48)	57.77 (+20.54)

Table 6: Maximum token norm in W8A8. Averaged over 500 images.

Model	Max token	
	Vanilla	w/ RegCache
CLIP	41.38	11.45
OpenCLIP	92.78	9.64
SigLIP	35.82	3.64
SigLIP2	148.20	15.16

Table 7: Ablation studies. PC: Prefix Caching; TD: Token Deleting.

Method	SigLIP-B/16	SigLIP2-B/16
Baseline	69.71	26.04
PC	74.37	23.82
TD	42.41	69.06
PC + TD	**74.38**	**72.35**

Generalizability of prefixes. Since the prefix search procedure in RegCache involves validation on the training split of the ImageNet-1k dataset, we additionally assess whether the learned prefixes generalize to other datasets, as the register token might have overfit to ImageNet-1k. In this spirit, we perform zero-shot classification on other datasets, with the results reported in Tab. 5. These results indicate that the prefix learned on ImageNet-1k remains effective on other datasets, suggesting that it acts as a transferable register token.

5.3 Analyses

Outlier reduction. Tab. 6 illustrates the change in the maximum token norm of the quantization-sensitive layer input when RegCache is applied. As expected, the maximum token norm decreases: This reduction effectively narrows the dynamic range of quantization, thereby improving quantization performance.

Ablation. To support our design choices, we ablate two components of RegCache: (1) prefix caching and (2) token deleting. In Tab. 7, we present the ablation study results for SigLIP and SigLIP2. The results show that the two stages, prefix caching and token deleting, act synergistically, yielding the best performance when both components are used together for both vision encoders. Surprisingly, when only one of the two steps is applied, we obtain even worse results than in the naïve case, further supporting the validity of our design choice.

Table 8: Latency results. We measure the latency (ms) of each model on a single NVIDIA A6000 GPU using TensorRT [32,33]. (a) Standalone vision encoder latency for CLIP-B/16 and SigLIP-B/16. (b) Vision encoder (VE) and language model (LM) prefill latency on Qwen3-VL-2B. We also report the end-to-end speedup (Accel.) achieved by quantizing only the vision encoder, while the LM quantized to INT8.

(a) Standalone VE. Batch size: 64.

Model	Method	Latency (ms)	Accel.
CLIP-B/16	Full-precision	132.13	–
	INT8	60.64	2.18×
	INT8 + RegCache	61.27 (+1.04%)	2.16×
SigLIP-B/16	Full-precision	128.49	–
	INT8	62.29	2.06×
	INT8 + RegCache	63.25 (+1.54%)	2.03×

(b) Qwen3-VL-2B

Data	Method	VE (ms)	LM Prefill (ms)	Accel.
Image	Full-precision	30.21	51.76	–
	INT8	8.33	11.63	2.10×
	INT8 + RegCache	8.40 (+0.80%)	11.81	2.08×
Video	Full-precision	79.18	124.00	–
	INT8	10.43	27.91	2.79×
	INT8 + RegCache	10.44 (+0.01%)	27.98	2.79×

5.4 Latency

We measure latency for both standalone vision encoders (Tab. 8a) and VLMs (Tab. 8b), and find that RegCache adds only marginal overhead (up to 1.54%). For VLMs, we profile time-to-first-token (TTFT): quantizing the vision encoder yields over a 2× TTFT reduction with the LLM quantized to INT8, indicating that the vision encoder becomes the primary bottleneck once the LLM is accelerated. We further observe that this acceleration is even more pronounced for video inputs, where processing multiple frames incurs substantial compute.

6 Conclusion

We introduce a training-free outlier mitigation algorithm, *RegCache*. RegCache can serve as an on-top method combined with existing PTQ algorithms for large-scale transformer-based vision encoders. Through extensive experiments, we demonstrate that RegCache consistently improves quantization performance across various tasks, indicating that it is indeed synergistic with other quantization methods. Our analyses reveal that RegCache suppresses activation outliers in quantization-sensitive layers, thereby narrowing the dynamic range of the input and improving quantization performance. Furthermore, we take a step toward identifying register tokens that are optimal for the quantization of vision encoders—a task that is inherently more elusive than for language models.

Limitations. A major limitation of our method is the need to tune several additional hyperparameters, such as the maximum number of tokens to delete and the number of prefix tokens. Another limitation is that the number of prefix and deleted tokens must be selected heuristically for each vision encoder and base quantization algorithm.

Discussions and future directions. Our work covers a variety of vision encoders, including those trained on multimodal data (e.g., CLIP) and those

trained on vision-only data (e.g., DINOv2). In our experiments, we observe that quantization-related measures (e.g., quantization sensitivity) behave somewhat differently across these cases, warranting further study. Another research direction arises from the differences between LLMs and ViT-based vision encoders: their outlier behavior differs significantly. Understanding this phenomenon would benefit a wide range of domains, including quantization and representation learning.

Acknowledgment

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00457882, No. RS-2019-II191906, No. RS-2022-II220713, No. RS-2026-25531289), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00453301, No. RS-2025-24873016, No. RS-2026-25494004), and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education.

References

1. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
2. Beyer, L., Steiner, A., Pinto, A.S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E., et al.: Paligemma: A versatile 3b vlm for transfer. arXiv preprint arXiv:2407.07726 (2024)
3. Bondarenko, Y., Nagel, M., Blankevoort, T.: Understanding and overcoming the challenges of efficient transformer quantization. In: Empirical Methods in Natural Language Processing (2021)
4. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101–mining discriminative components with random forests. In: European conference on computer vision (2014)
5. Chen, M., Liu, Y., Wang, J., Bin, Y., Shao, W., Luo, P.: PrefixQuant: Eliminating outliers by prefixed tokens for large language models quantization. arXiv preprint arXiv:2410.05265 (2024)
6. Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J.: Reproducible scaling laws for contrastive language-image learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2023)
7. Choukroun, Y., Kravchik, E., Yang, F., Kisilev, P.: Low-bit quantization of neural networks for efficient inference. In: IEEE/CVF International Conference on Computer Vision Workshop (2019)
8. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., , Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2014)
9. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers. In: International Conference on Learning Representations (2024)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition (2009)
11. Dettmers, T., Lewis, M., Belkada, Y., Zettlemoyer, L.: GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In: Advances in neural information processing systems (2022)
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
13. Faghri, F., Vasu, P.K.A., Koc, C., Shankar, V., Toshev, A.T., Tuzel, O., Pouransari, H.: MobileCLIP2: Improving multi-modal reinforced training. Transactions on Machine Learning Research (2025)
14. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: conference on computer vision and pattern recognition workshop (2004)
15. Fu, C., Dai, Y., Luo, Y., Li, L., Ren, S., Zhang, R., Wang, Z., Zhou, C., Shen, Y., Zhang, M., et al.: Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In: CVPR (2025)
16. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)

17. Guo, Z., Kamigaito, H., Watanabe, T.: Attention score is not all you need for token importance indicator in kv cache reduction: Value also matters. In: Empirical Methods in Natural Language Processing (2024)
18. Hudson, D.A., Manning, C.D.: GQA: A new dataset for real-world visual reasoning and compositional question answering. Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
19. Jiang, N., Dravid, A., Efros, A.A., Gandelsman, Y.: Vision transformers don't need trained registers. In: Advances in neural information processing systems (2025)
20. Kang, S., Kim, J., Kim, J., Hwang, S.J.: See what you are told: Visual attention sink in large multimodal models. In: International Conference on Learning Representations (2025)
21. Kovaleva, O., Kulshreshtha, S., Rogers, A., Rumshisky, A.: BERT busters: Outlier dimensions that disrupt transformers. In: Findings of the ACL (2021)
22. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE international conference on computer vision workshops (2013)
23. Li, Y., Wang, C., Jia, J.: Llama-vid: An image is worth 2 tokens in large language models. In: European Conference on Computer Vision (2024)
24. Li, Z., Xiao, J., Yang, L., Gu, Q.: RepQ-ViT: Scale reparameterization for post-training quantization of vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023)
25. Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.M., Wang, W.C., Xiao, G., Dang, X., Gan, C., Han, S.: AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In: Proceedings of Machine Learning and Systems (2024)
26. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision (2014)
27. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems (2023)
28. Liu, Y., Yang, H., Dong, Z., Keutzer, K., Du, L., Zhang, S.: NoisyQuant: Noisy bias-enhanced post-training activation quantization for vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023)
29. Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., Gao, W.: Post-training quantization for vision transformer. Advances in Neural Information Processing Systems (2021)
30. Lu, A., Liao, W., Wang, L., Yang, H., Shi, J.: Artifacts and attention sinks: Structured approximations for efficient vision transformers. arXiv preprint arXiv:2507.16018 (2025)
31. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Indian conference on computer vision, graphics & image processing (2008)
32. NVIDIA Corporation: Deploying hugging face llava1.5-7b model in triton. https://docs.nvidia.com/deeplearning/triton-inference-server/user-guide/docs/tutorials/Popular_Models_Guide/Llava1.5/llava_trtllm_guide.html, accessed February 2026
33. NVIDIA Corporation: Nvidia tensorrt. https://developer.nvidia.com/tensorrt, accessed February 2026
34. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N.,

Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning robust visual features without supervision. Transactions on Machine Learning Research (2024)
35. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)
36. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning (2021)
37. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al.: Dinov3. arXiv preprint arXiv:2508.10104 (2025)
38. Son, S., Park, W., Han, W., Kim, K., Lee, J.: Prefixing attention sinks can mitigate activation outliers for large language model quantization. In: Empirical Methods in Natural Language Processing (2024)
39. Sun, M., Chen, X., Kolter, J.Z., Liu, Z.: Massive activations in large language models. In: Conference on Language Modeling (2024)
40. Timkey, W., Van Schijndel, M.: All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In: Empirical Methods in Natural Language Processing (2021)
41. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., et al.: SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. arXiv preprint arXiv:2502.14786 (2025)
42. Vasu, P.K.A., Faghri, F., Li, C.L., Koc, C., True, N., Antony, A., Santhanam, G., Gabriel, J., Grasch, P., Tuzel, O., et al.: Fastvlm: Efficient vision encoding for vision language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference (2025)
43. Wu, Z., Wang, S., Zhang, J., Chen, J., Wang, Y.: FIMA-Q: Post-training quantization for vision transformers by fisher information matrix approximation. In: Proceedings of the Computer Vision and Pattern Recognition Conference (2025)
44. Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., Han, S.: SmoothQuant: Accurate and efficient post-training quantization for large language models. In: International conference on Machine Learning (2023)
45. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient streaming language models with attention sinks. In: International Conference on Learning Representations (2024)
46. Xiao, K.Y., Engstrom, L., Ilyas, A., Madry, A.: Noise or signal: The role of image backgrounds in object recognition. In: International Conference on Learning Representations (2021)
47. Yang, J., Kim, H., Kim, Y.: Mitigating quantization errors due to activation spikes in GLU-based LLMs. arXiv preprint 2405.14428 (2024)
48. Yuan, Z., Xue, C., Chen, Y., Wu, Q., Sun, G.: PTQ4ViT: Post-training quantization for vision transformers with twin uniform quantization. In: European conference on computer vision (2022)
49. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF international conference on computer vision (2023)

50. Zhong, Y., Huang, Y., Hu, J., Zhang, Y., Ji, R.: Towards accurate post-training quantization of vision transformers via error reduction. IEEE Trans. Pattern Anal. Mach. Intell. (2025)
51. Zhou, J., Shu, Y., Zhao, B., Wu, B., Liang, Z., Xiao, S., Qin, M., Yang, X., Xiong, Y., Zhang, B., et al.: MLVU: Benchmarking multi-task long video understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2025)