Activation Quantization of Vision Encoders Needs Prefixing Registers

Seunghyeon Kim [1] **Taesun Yeom** [1] **Jinho Kim** [2] **Wonpyo Park** [3] **Kyuyeun Kim** [3]

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 activation 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).

1. Introduction

Transformer-based vision encoders, such as CLIP and DINOv2, are core components of modern multimodal systems and are often deployed as standalone models on edge devices, where storage constraints and inference overhead become major bottlenecks (Faghri et al., 2025). [1] Vision encoders also serve as the visual backbones of many vision-language models (VLMs) (Liu et al., 2023a; Beyer et al., 2024), where their computational cost remains substantial, especially for high-resolution images or video (Li et al., 2024; Vasu et al., 2025). 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 (Li et al., 2024; Vasu et al., 2025).

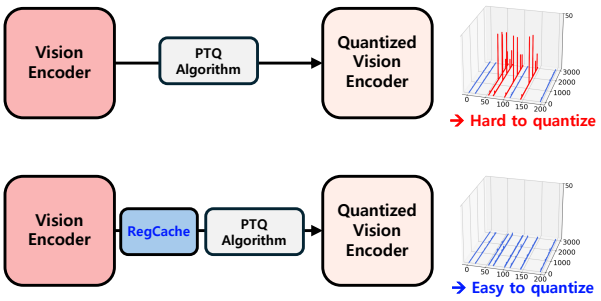

Figure 1. A diagram of the proposed RegCache framework, preprocessing the vision encoder before PTQ.

To address these challenges, post-training quantization (PTQ) offers a practical solution, reducing memory usage and computational cost without additional training (Choukroun et al., 2019). However, despite recent progress, PTQ methods for ViT-based vision encoders still suffer from significant performance degradation under low-bit quantization (Li et al., 2023; Wu et al., 2025; Zhong et al., 2025).

In particular, 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 (Sun et al., 2024). These outliers significantly expand the activation quantization range, leading to large quantization errors. While outlier-robust quantization has been widely studied for LLMs (Dettmers et al., 2022; Xiao et al., 2023; Lin et al., 2024), 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 (Son et al., 2024; Chen et al., 2024).

An emerging alternative is to directly mitigate outliers by

[1]POSTECH, Pohang, Korea [2]Dankook University, Yongin, Korea [3]Google, California, USA. Correspondence to: Jaeho Lee <jaeho.lee@postech.ac.kr>.

Proceedings of the Workshop on Resource-Adaptive Foundation Model Inference at the 43^{rd} International Conference on Machine Learning, Seoul, South Korea, 2026.

[1]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).

prefixing *attention sink* tokens—i.e., semantically meaningless tokens such as ⟨BOS⟩ or ⟨SEP⟩ that absorb large attention from other tokens (Xiao et al., 2024; Sun et al., 2024). 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 (Yang et al., 2024; Son et al., 2024; Chen et al., 2024).

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. Recent work suggests that introducing such meaningless tokens—termed *registers*—during training can improve the interpretability of ViT-based models (Darcet et al., 2024). However, incorporating registers remains uncommon in vision encoders [2], and importantly, they cannot reduce the activation quantization range.

Contribution. We introduce ***RegCache*** (Register Caching), a training-free prefix-based outlier mitigation algorithm for quantizing pretrained vision encoders. We observe that sink tokens emerge from **middle layers** and exhibit **high similarity across images**, enabling their use as a universal register at test time.

Based on this, RegCache prefixes precomputed middle-layer registers as a key-value (KV) cache and removes residual sink tokens with large activations, preventing activation range inflation and improving quantization stability. Notably, RegCache requires no retraining and can be easily integrated into existing PTQ pipelines as a lightweight on-top module.

Across diverse vision encoders and PTQ methods, RegCache consistently improves accuracy, with the largest gains observed in low-bit settings (e.g., 4-bit).

2. Related work

Outliers and attention sink tokens in transformer-based vision encoders. In ViTs, outlier tokens typically correspond to uninformative background patches, and attention sink tokens—tokens with little or no semantic content that nonetheless attract excessive attention—are closely related to the emergence of such outliers (Darcet et al., 2024; Xiao et al., 2024; Guo et al., 2024; Jiang et al., 2025; Lu et al., 2025). These sink tokens act as noise in the attention map, hindering patch interactions and degrading downstream visual performance (Darcet et al., 2024; Jiang et al., 2025; Kang et al., 2025; Lu et al., 2025). Prior work mitigates this issue by introducing additional register tokens during training or inference (Darcet et al., 2024; Jiang et al., 2025). From a different perspective, we instead leverage sink tokens to suppress outliers that degrade PTQ performance. Our method employs a precomputed sink token, enabling efficient inference without per-image processing while reducing the activation dynamic range prior to 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 (Liu et al., 2021). 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 (Li et al., 2023) and PTQ4ViT (Yuan et al., 2022) propose quantization schemes that isolate and minimize the impact of outliers. Other approaches address the heavy-tailed activation distribution; for example, NoisyQuant (Liu et al., 2023b) injects noise to reshape the activation distribution. FIMA-Q (Wu et al., 2025) introduces round-function optimization for PTQ, while ERQ (Zhong et al., 2025) proposes a two-step procedure that sequentially quantizes activations and weights to reduce quantization error. While existing PTQ methods mitigate quantization errors through specialized quantizers and distribution reshaping, RegCache directly suppresses outliers via structural modification.

3. A closer look at outliers in vision encoders

Outliers in vision encoders tend to emerge at seemingly random background patch tokens for a given image, whereas in LLMs they typically occur at specific token positions or types (Sun et al., 2024; Darcet et al., 2024; Son et al., 2024). This lack of consistency makes it difficult to mitigate outliers in vision encoders through prefixing. In this section, we present two key findings that motivate an alternative strategy—*identifying universal sink tokens in the middle blocks of the model and prefixing them there*.

3.1. Layerwise quantization sensitivity and outliers

We first analyze the quantization sensitivity of each layer in vision encoders and its connection to the emergence of activation outliers (i.e., FC2 inputs). In Figure 2 (top), we report layerwise quantization sensitivity, measured by zero-shot ImageNet-1k accuracy when each layer is quantized to RTN W8A8. We observe that quantization-sensitive layers—i.e., layers that incur substantial accuracy drops when quantized—are largely 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. As shown in Figure 2

[2] In this regard, DINOv3 is a pleasant exception (Siméoni et al., 2025)

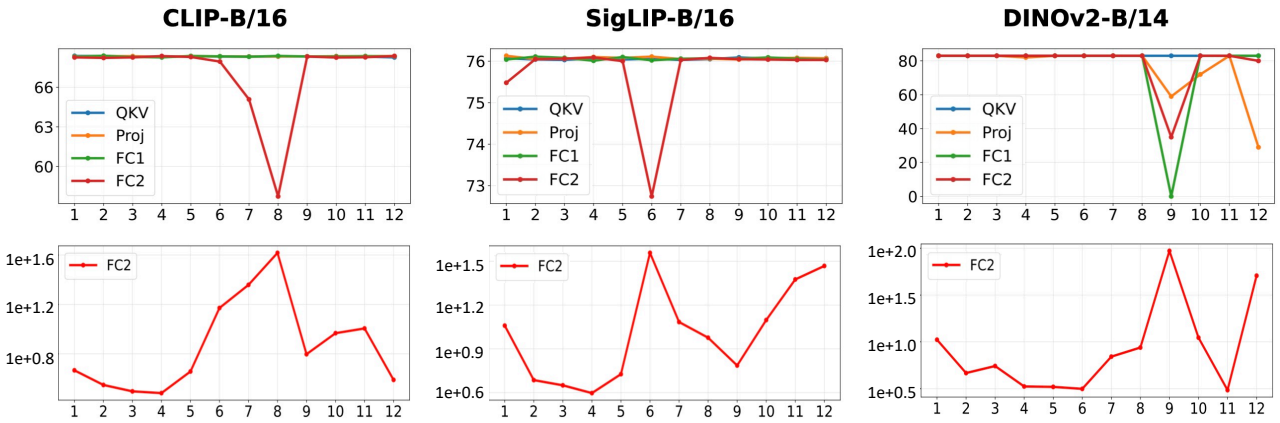

Figure 2. **(Top) Layerwise quantization sensitivity (%).** We plot the zero-shot ImageNet-1k accuracy when quantizing each layer individually to W8A8. **(Bottom) Maximum norm of the FC2 layer input tokens for each layer.** We plot the largest ℓ_∞-norm across all tokens per image on a log scale, averaged over the ImageNet-1k validation set. For both plots, the x-axis denotes the transformer block index.

(bottom), these layers coincide with the blocks where activation outliers in the hidden states begin to emerge, suggesting that outliers are a primary driver of performance degradation in quantized vision encoders.

In particular, we establish a clear connection between quantized accuracy and high-norm activation behavior, suggesting 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 Section A. We further explore why outliers emerge in the middle layers, in contrast to LLMs in Section B.

3.2. Universality 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.

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

Token Type	Cosine sim.
Normal tokens	0.26 ($\pm$0.10)
Outlier tokens	0.89 ($\pm$0.07)

From Table 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 only 0.26. This suggests that outliers contain components largely independent of the input image and may represent universal features that persist across samples. In Section G, we provide a theoretical explanation for this phenomenon, showing that it arises from the large magnitude of the outliers and the alignment of their locations (i.e., channels).

4. Method

Before introducing our method, we recall two observations from Section 3: (i) In vision encoders, outliers tend to emerge in the middle layers. (ii) 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 (Son et al., 2024)—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. RegCache follows three steps: curating, caching, and deleting (see Figure 3).

4.1. Curating

Given a pretrained vision encoder, we first identify its quantization-sensitive layer and then construct a set of register candidates by selecting the top-k tokens with the largest activation in that layer from a pool of reference images.

Identifying the quantization-sensitive layer. Following Section 3.1, we quantize each layer independently and select the one that yields the largest accuracy drop on a reference task. As the reference task, we use ImageNet-1k classification on the training split. As a label-free alternative, we

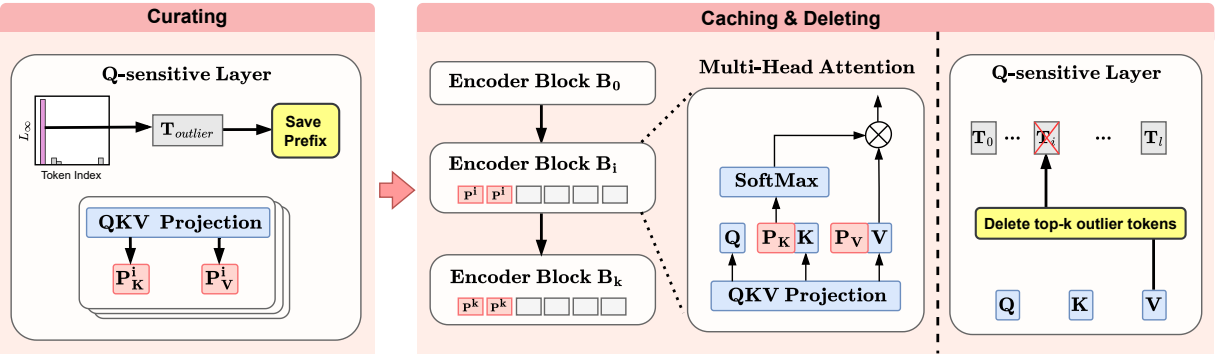

Figure 3. **An overview of the proposed RegCache algorithm.** We identify a universal register by analyzing the inputs of quantization-sensitive layers across blocks. During inference, the register is inserted into each block, and outlier tokens are removed from the most quantization-sensitive layer.

also provide a reconstruction-loss-based approach using unlabeled data (see Section L).

Curating the set of register candidates. After identifying the quantization-sensitive layer, we run inference on a pool of reference images and select the tokens with the largest ℓ_∞ norm at that layer. Let l_{q} denote the quantization-sensitive layer, 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

$$\mathcal{S} = \operatorname{argtopk}\left\{\|\mathbf{z}\|_\infty \;\middle|\; \mathbf{z} \in \Phi_{l_q}(\mathbf{x}),\ \mathbf{x} \in \mathcal{I}_{\mathrm{ref}}\right\}, \qquad (1)$$

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 l_{q}, we repeat the same search for up to three preceding blocks.

4.2. Caching

Having constructed the register candidate set $\mathcal{S}$, we compose the register by 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:

- We first compute KV caches for each register candidate token $\mathbf{z} \in \mathcal{S}$, for blocks starting from a few layers before the quantization-sensitive block as well as all subsequent blocks, using the *unquantized* vision encoder.
- We then insert the averaged KV cache into the *quantized* vision encoder with different numbers of copies. We vary τ within $\{1, 2, \ldots, 15\}$ and select τ^* as the one with the highest reference task accuracy.[3] As in Section 4.1, we use classification on the training split of ImageNet-1k as the reference task.

4.3. Deleting

Finally, we apply a *token deletion* process to the input of the quantization-sensitive block. At inference time, this removes sink tokens that emerge among the image patch tokens, thereby removing any remaining outliers. Given a test image $\mathbf{x}_{\mathrm{test}}$, we select the tokens with the top-$\tilde{k}$ ℓ_∞ norm,

$$\mathcal{D} = \operatorname{argtopk}\tilde{}\left\{\|\mathbf{z}\|_\infty \;\middle|\; \mathbf{z} \in \Phi_{l_q}(\mathbf{x}_{\mathrm{test}})\right\}, \qquad (2)$$

and remove them from the model. As in the curating and caching steps, the number of removed tokens, *i.e.*, $\tilde{k}$, is tuned using the reference task.

RegCache requires no additional training and only several rounds of validation on a reference task and dataset. As a result, it does not require large amounts of data or substantial computational resources. At inference time, RegCache temporarily adds and removes a few tokens, introducing only negligible overhead (see Section 5.4).

5. Experiments

5.1. Setup

Standalone vision encoders. We evaluate the proposed method on five widely used vision encoders: CLIP, OpenCLIP, SigLIP, SigLIP2 and DINOv2, which cover diverse training configurations. For evaluation, we measure the quality of the quantized vision encoders using zero-shot image classification accuracy on ImageNet-1k (Deng et al., 2009). We further evaluate image retrieval accuracy on MS-COCO dataset in Section D.

Vision-language models. We primarily use Qwen3-VL (Bai et al., 2025) (2B and 8B), a strong open-source VLM

[3] 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.

Method	CLIP-B/16			OpenCLIP-B/16			SigLIP-B/16			SigLIP2-B/16			DINOv2-B/14		
	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

Table 3. **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)

for both image and video understanding. To demonstrate the broad applicability of our method, we also evaluate it on other VLMs, including LLaVA family (Liu et al., 2023a) (see Section E). We evaluate image understanding on GQA (Hudson & Manning, 2019) and VQAv2 (Goyal et al., 2017), and video understanding on Video-MME (Fu et al., 2025) and MLVU (Zhou et al., 2025), including long-video settings where vision-encoder overhead can be a bottleneck.

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 (Yuan et al., 2022), RepQ-ViT (Li et al., 2023), and FIMA-Q (Wu et al., 2025)), (2) rounding function optimization (ERQ (Zhong et al., 2025) and FIMA-Q (Wu et al., 2025)), (3) weight correction (ERQ (Zhong et al., 2025)), and (4) activation distribution shaping (NoisyQuant (Liu et al., 2023b)) to reduce quantization error. For further details, see Section F.

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.

5.2. Main results

Standalone vision encoders. In Table 2, we report zero-shot ImageNet-1k classification accuracy. We observe that baselines combined with RegCache consistently achieve better accuracy in most settings (see Section C for results on other datasets). Specifically, they outperform the base quantization methods in both best accuracy gap (Best Δ) and average accuracy gap (Average Δ). Only one setup—DINOv2—shows a negligible accuracy drop.

Vision-language models. We further evaluate our quantized vision encoders within VLMs on image and video benchmarks (Table 3). 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 Table 3, RegCache consistently improves accuracy in VLMs, demonstrating robust gains across benchmarks.

Table 4. **Latency results.** We measure the latency (ms) of each model on a single NVIDIA A6000 GPU using TensorRT (NVIDIA Corporation, a;b). (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 is 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×

Table 5. **Reduction in maximum token norm within the input of quantization-sensitive layers in W8A8.** We report the mean across 500 image samples.

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

Table 6. **Ablation studies.** We compare the contribution of each component of RegCache on SigLIP-B/16 and SigLIP2-B/16. Caching and deleting play a complementary role, enabling a wide coverage over a wide range of vision encoders.

Method	SigLIP-B/16	SigLIP2-B/16
Baseline	69.71	26.04
Prefix Caching	74.37	23.82
Token Deleting	42.41	69.06
Prefix Caching + Token Deleting	**74.38**	**72.35**

5.3. Analyses

Outlier reduction. Table 5 shows how the maximum token norm of the input to the quantization-sensitive layer changes when RegCache is applied. RegCache reduces the maximum token norm, thereby narrowing the quantization dynamic range and improving quantization performance.

Ablation. To support our design choices, we ablate the two components of RegCache: prefix caching and token deleting. In Table 6, we report results for SigLIP and SigLIP2. The two components act synergistically, achieving the best performance when used together. 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.

5.4. Latency

We measure latency for both standalone vision encoders (Table 4a) and VLMs (Table 4b), 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, showing that the vision encoder becomes the primary bottleneck once the LLM is accelerated. The acceleration is even more pronounced for video inputs, where multiple frames incur substantial compute.

6. Conclusion

In this paper, we introduce *RegCache*, a training-free outlier mitigation algorithm for large-scale transformer-based vision encoders. RegCache serves as a plug-in method and can be combined with existing PTQ algorithms. Through extensive experiments, we show that it consistently improves quantization performance across various tasks and works synergistically with other quantization methods. Our analyses reveal that RegCache suppresses activation outliers in quantization-sensitive layers, narrowing the input dynamic range and improving quantization performance. Furthermore, we take a step toward identifying register tokens that are optimal for vision encoder quantization—a task more elusive than in language models.

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

Discussion and future directions. Our work covers a variety of vision encoders, including those trained on multimodal data (e.g., CLIP) and vision-only data (e.g., DINOv2). In our experiments, quantization-related measures (e.g., quantization sensitivity) behave somewhat differently

across these cases, warranting further study. Another direction stems from the differences between LLMs and vision encoders, whose outlier behavior differs significantly (see Section N for an extended discussion). Understanding this phenomenon could benefit a wide range of domains, including quantization and representation learning.

References

Ashkboos, S., Mohtashami, A., Croci, M. L., Li, B., Cameron, P., Jaggi, M., Alistarh, D., Hoefler, T., and Hensman, J. QuaRot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 2024.

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.

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.

Bolya, D., Huang, P.-Y., Sun, P., Cho, J. H., Madotto, A., Wei, C., Ma, T., Zhi, J., Rajasegaran, J., Rasheed, H. A., Wang, J., Monteiro, M., Xu, H., Dong, S., Ravi, N., Li, S.-W., Dollar, P., and Feichtenhofer, C. Perception encoder: The best visual embeddings are not at the output of the network. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Chen, M., Liu, Y., Wang, J., Bin, Y., Shao, W., and Luo, P. PrefixQuant: Eliminating outliers by prefixed tokens for large language models quantization. *arXiv preprint arXiv:2410.05265*, 2024.

Choukroun, Y., Kravchik, E., Yang, F., and Kisilev, P. Low-bit quantization of neural networks for efficient inference. In *IEEE/CVF International Conference on Computer Vision Workshop*, 2019.

Darcet, T., Oquab, M., Mairal, J., and Bojanowski, P. Vision transformers need registers. In *International Conference on Learning Representations*, 2024.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, 2009.

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in neural information processing systems*, 2022.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Faghri, F., Vasu, P. K. A., Koc, C., Shankar, V., Toshev, A. T., Tuzel, O., and Pouransari, H. MobileCLIP2: Improving multi-modal reinforced training. *Transactions on Machine Learning Research*, 2025.

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.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and 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.

Gu, X., Pang, T., Du, C., Liu, Q., Zhang, F., Du, C., Wang, Y., and Lin, M. When attention sink emerges in language models: An empirical view. In *International Conference on Learning Representations*, 2025.

Guo, Z., Kamigaito, H., and 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.

Hudson, D. A. and 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.

Jiang, N., Dravid, A., Efros, A. A., and Gandelsman, Y. Vision transformers don't need trained registers. In *Advances in neural information processing systems*, 2025.

Kang, S., Kim, J., Kim, J., and Hwang, S. J. See what you are told: Visual attention sink in large multimodal models. In *International Conference on Learning Representations*, 2025.

Li, Y., Wang, C., and Jia, J. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, 2024.

Li, Z., Xiao, J., Yang, L., and 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.

Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.-M., Wang, W.-C., Xiao, G., Dang, X., Gan, C., and Han, S. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, 2024.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. *Advances in neural information processing systems*, 2023a.

Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024a.

Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., and Lee, Y. J. Llavanext: Improved reasoning, ocr, and world knowledge, 2024b.

Liu, Y., Yang, H., Dong, Z., Keutzer, K., Du, L., and 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*, 2023b.

Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., and Gao, W. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 2021.

Liu, Z., Zhao, C., Fedorov, I., Soran, B., Choudhary, D., Krishnamoorthi, R., Chandra, V., Tian, Y., and Blankevoort, T. SpinQuant: LLM quantization with learned rotations. In *International Conference on Learning Representations*, 2025.

Lu, A., Liao, W., Wang, L., Yang, H., and Shi, J. Artifacts and attention sinks: Structured approximations for efficient vision transformers. *arXiv preprint arXiv:2507.16018*, 2025.

NVIDIA Corporation. Nvidia tensorrt. https://developer.nvidia.com/tensorrt, a.

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, b.

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.

Son, S., Park, W., Han, W., Kim, K., and Lee, J. Prefixing attention sinks can mitigate activation outliers for large language model quantization. In *Empirical Methods in Natural Language Processing*, 2024.

Sun, M., Chen, X., Kolter, J. Z., and Liu, Z. Massive activations in large language models. In *Conference on Language Modeling*, 2024.

Tai, Y.-S. et al. MPTQ-ViT: Mixed-precision post-training quantization for vision transformer. *arXiv preprint arXiv:2401.14895*, 2024.

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.

Wu, Z., Wang, S., Zhang, J., Chen, J., and 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.

Xiao, G., Lin, J., Seznec, M., Wu, H., Demouth, J., and Han, S. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International conference on Machine Learning*, 2023.

Xiao, G., Tian, Y., Chen, B., Han, S., and Lewis, M. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, 2024.

Xiao, K. Y., Engstrom, L., Ilyas, A., and Madry, A. Noise or signal: The role of image backgrounds in object recognition. In *International Conference on Learning Representations*, 2021.

Yang, J., Kim, H., and Kim, Y. Mitigating quantization errors due to activation spikes in GLU-based LLMs. *arXiv preprint 2405.14428*, 2024.

Yuan, Z., Xue, C., Chen, Y., Wu, Q., and Sun, G. PTQ4ViT: Post-training quantization for vision transformers with twin uniform quantization. In *European conference on computer vision*, 2022.

Zhao, T., Fang, T., Huang, H., Wan, R., Soedarmadji, W., Liu, E., Li, S., Lin, Z., Dai, G., Yan, S., Yang, H., Ning, X., and Wang, Y. ViDiT-Q: Efficient and accurate quantization of diffusion transformers for image and video generation. In *International Conference on Learning Representations*, 2025.

Zhong, Y., Huang, Y., Hu, J., Zhang, Y., and Ji, R. Towards accurate post-training quantization of vision transformers via error reduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2025.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

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.

Appendix

A. Additional Results on Quantization Sensitivity

In this section, we provide additional plots for other vision encoders, analogous to those in Figure 2. In Figure 4, we plot layerwise quantization sensitivity (top row) and maximum token norm (bottom row) for OpenCLIP and SigLIP2. The trends are consistent with our analysis in Section 3.1: increases in maximum token norm coincide with decreases in quantization accuracy. However, in the case of SigLIP2, the absolute scale of the maximum norm is significantly larger than in the other architectures we considered. Consequently, applying RegCache yields a clearer benefit, as shown in Table 2. Given SigLIP2's distinct behavior compared to other vision encoders, it would be intriguing to investigate further; we leave this for future work.

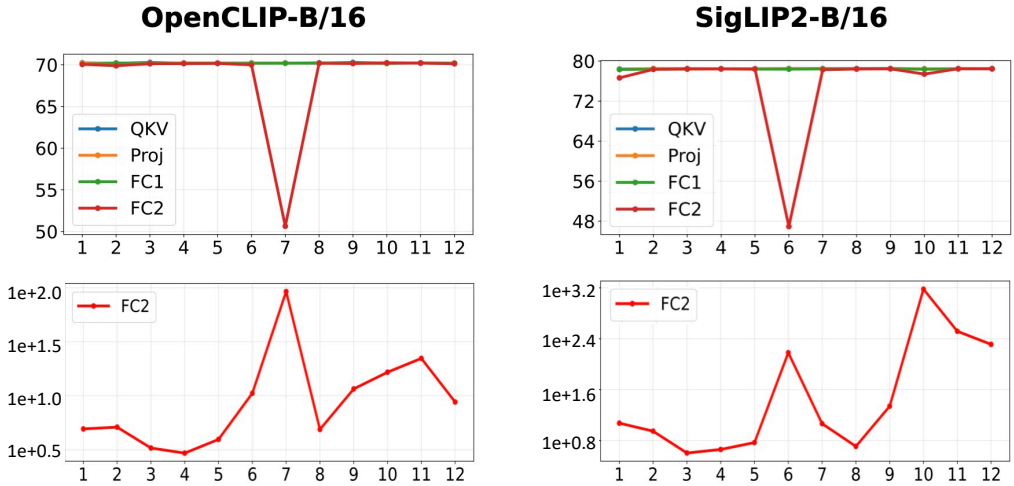

Figure 4. **(Top) Layerwise quantization sensitivity (%).** Zero-shot ImageNet-1k accuracy when we quantize only one layer to W8A8. **(Bottom) Layerwise max token norms.** The largest ℓ_∞-norm of all tokens in an image on a logarithmic scale, averaged over the ImageNet-1k validation set.

B. Why the middle layers?

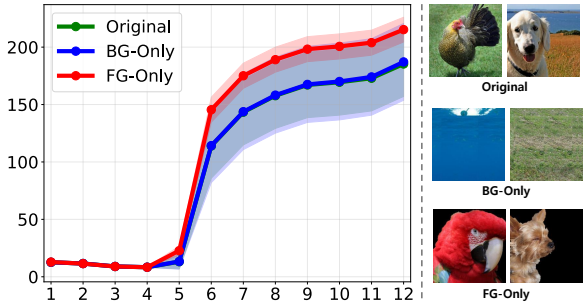

Figure 5. **Outliers in FG/BG-only images, for SigLIP-B/16 model.** Note that "Original" and "BG-only" nearly overlap.

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⟩.

To test this hypothesis, we design an experiment that compares the emergence of outliers in images where meaningless patches are easily identifiable against those where the distinction is less clear. Specifically, using the test set of ImageNet-9 (Xiao et al., 2021), we compare foreground-only images—where the background pixels are zero-ed out—with the original images. In foreground-only images, semantically meaningless patches should be easier to identify, requiring fewer processing blocks.

As shown in Figure 5, outliers in foreground-only images 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 Section N, 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.

C. Generalizability

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 overfitted to ImageNet-1k. In this spirit, we perform zero-shot classification on other datasets, with the results reported in Table 7. These results indicate that the prefix learned on ImageNet-1k remains effective on other datasets, suggesting that it acts as a universal register token.

Table 7. Zero-shot classification accuracy (%) on various datasets.

Model	Method	StanfordCars	Flowers-102	Food-101	CIFAR-100	Caltech101	Oxford-III Pet	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)

D. Additional Results on Image Retrieval

To further evaluate performance, we present zero-shot image–text retrieval results on the MS-COCO dataset in Table 8. Across all models, RegCache consistently achieves the highest retrieval accuracy.

Table 8. **Zero-shot image–text retrieval performance on MS-COCO (R@1).** The table is organized into two mini-tables: **(top) CLIP/OpenCLIP** and **(bottom) SigLIP/SigLIP2**. The best quantized 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.

(a) CLIP-B/16 and OpenCLIP-B/16

	CLIP-B/16						OpenCLIP-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	61.02	61.02	61.02	41.38	41.38	41.38
Naïve	0.00	0.00	22.76	0.01	0.06	14.08	0.02	0.16	37.32	0.03	0.24	26.30
w/ RegCache	0.02	0.22	46.10	0.04	0.44	28.02	0.00	0.32	57.60	0.04	0.77	38.45
PTQ4ViT	0.06	37.28	52.78	0.17	23.00	32.00	0.04	50.60	59.60	0.06	34.56	40.66
w/ RegCache	0.52	43.46	53.54	0.14	26.51	32.48	1.00	**60.00**	**60.00**	3.00	**40.96**	**40.96**
RepQ-ViT	0.32	29.06	44.52	0.47	15.90	23.01	0.12	17.60	57.62	0.21	8.90	38.88
w/ RegCache	4.32	38.68	45.58	3.37	19.70	23.68	1.56	32.96	59.44	1.38	14.90	39.82
NoisyQuant	0.12	25.26	48.94	0.19	18.02	31.07	0.46	43.86	53.84	1.20	27.05	34.50
w/ RegCache	0.37	33.86	49.10	0.71	22.28	30.40	2.18	51.58	56.06	2.81	31.73	35.53
FIMA-Q	40.92	52.90	53.58	27.71	32.33	32.76	55.82	58.84	59.26	38.02	40.39	40.21
w/ RegCache	**51.82**	**53.22**	**53.66**	**33.43**	**32.88**	**33.01**	**57.34**	59.88	59.86	**40.02**	40.30	40.58
ERQ	0.32	23.62	44.78	3.71	15.58	23.31	9.70	32.56	58.74	4.28	14.74	39.83
w/ RegCache	13.78	40.88	44.86	6.52	19.82	23.52	13.02	33.64	59.56	5.19	15.38	40.04
Best Δ	+13.46	+17.26	+1.06	+5.72	+4.26	+0.67	+3.32	+15.36	+2.22	+2.94	+6.40	+1.03
Average Δ	+5.81	+8.40	+0.43	+2.38	+3.27	+0.19	+1.79	+6.92	+1.17	+1.73	+3.53	+0.57

(b) SigLIP-B/16 and SigLIP2-B/16

	SigLIP-B/16						SigLIP2-B/16					
	Image → Text			Text → Image			Image → Text			Text → Image		
	W4A4	W6A6	W8A8	W4A4	W6A6	W8A8	W4A4	W6A6	W8A8	W4A4	W6A6	W8A8
FP32	67.68	67.68	67.68	47.19	47.19	47.19	71.60	71.60	71.60	52.33	52.33	52.33
Naïve	0.00	0.34	60.04	0.04	0.86	41.80	0.02	0.02	14.26	0.02	0.16	13.86
w/ RegCache	0.04	12.06	65.76	0.04	12.70	46.30	0.02	0.08	64.02	0.03	0.22	46.80
PTQ4ViT	0.20	60.66	66.86	0.78	41.73	47.16	0.04	29.88	69.74	0.19	27.33	51.62
w/ RegCache	2.10	61.42	67.72	1.78	43.05	47.61	0.70	53.90	70.12	2.64	39.96	51.68
RepQ-ViT	5.14	37.50	65.90	6.21	26.41	46.33	0.16	57.00	69.20	0.28	39.20	50.15
w/ RegCache	14.92	61.06	66.12	9.37	43.63	46.42	3.20	58.78	69.76	3.49	41.26	50.15
NoisyQuant	0.29	52.52	67.10	1.18	33.36	46.76	0.07	28.74	62.04	0.65	25.28	46.32
w/ RegCache	2.53	63.28	67.24	3.81	43.25	**47.64**	0.32	50.66	70.24	1.79	39.25	51.41
FIMA-Q	67.66	67.70	67.62	47.23	47.22	47.21	67.70	67.68	67.70	47.22	47.21	47.22
w/ RegCache	**68.14**	**68.14**	**68.18**	**47.64**	**47.62**	47.62	**71.54**	**71.54**	**71.52**	**52.49**	**52.49**	**52.50**
ERQ	28.96	61.94	65.80	20.70	43.48	46.46	8.28	64.80	69.76	5.78	45.44	49.97
w/ RegCache	32.66	62.30	66.16	23.21	44.61	46.63	12.30	65.56	70.00	9.18	46.28	50.20
Best Δ	+9.78	+23.56	+0.86	+3.16	+17.22	+0.88	+4.02	+24.02	+8.20	+5.27	+13.97	+5.28
Average Δ	+3.62	+7.18	+0.43	+1.94	+5.99	+0.40	+2.36	+10.47	+2.64	+3.09	+6.96	+2.13

E. Additional Results on VLMs

We further evaluate RegCache on the widely adopted LLaVA family (Liu et al., 2024a;b) on the GQA and VQAv2 benchmarks, as summarized in Table 9. In Table 3, we consider a practical deployment setting with 4-bit quantization and RepQ-ViT as the PTQ baseline. We note that RepQ-ViT is particularly effective in practice, offering both favorable accuracy gains and low calibration overhead, especially for larger vision encoders in VLMs. Consistent with our observations on Qwen3-VL (Bai et al., 2025), RegCache delivers stable gains in this setting, confirming that its efficacy extends beyond a specific architecture.

Table 9. **VLM performance on LLaVA architecture-based models under 4-bit quantization.**

(a) **LLaVA-1.5 models.**

Method	LLaVA-1.5-7B		LLaVA-1.5-13B	
	GQA	VQAv2	GQA	VQAv2
FP32	60.78	76.64	62.57	78.29
Naïve	33.84	35.62	34.70	35.89
RepQ-ViT	41.40	45.87	43.27	47.85
w/ RegCache	45.10 (+3.70)	50.11 (+4.24)	46.21 (+2.94)	52.15 (+4.30)

(b) **LLaVA-NeXT models.**

Method	LLaVA-1.6-vicuna-13B		LLaVA-1.6-mistral-7B	
	GQA	VQAv2	GQA	VQAv2
FP32	63.08	79.33	56.59	76.15
Naïve	35.72	36.70	31.87	35.21
RepQ-ViT	48.93	56.70	44.63	53.45
w/ RegCache	51.40 (+2.47)	58.83 (+2.13)	46.21 (+1.58)	55.64 (+2.19)

F. Experimental Settings and Baselines

In this section, we provide additional details on the vision encoder models, quantization baselines, and the hardware setup used in our experiments.

Models. The vision encoders we choose cover a broad range of training objectives (image–text contrastive learning vs. self-supervised learning) and architectural choices for global feature aggregation, providing a diverse testbed for evaluating RegCache. CLIP and SigLIP are trained on image–text pairs with contrastive objectives, whereas DINOv2 is trained on image-only datasets. Regarding input tokens, CLIP and DINOv2 utilize a class token to extract global features, while SigLIP and SigLIP2 use patch-wise pooling to generate a token that captures global information.

Baselines. We consider the following baselines ranging from widely used methods to recent ones:

- **PTQ4ViT** (Yuan et al., 2022) proposes a twin uniform quantizer to handle the unbalanced activation distributions found in ViTs, particularly after non-linearities such as Softmax and GELU.
- **RepQ-ViT** (Li et al., 2023) addresses quantization bottlenecks by applying specialized preprocessing to sensitive layers, such as channel-wise quantization after LayerNorm and log2 quantization after Softmax.
- **NoisyQuant** (Liu et al., 2023b) introduces a quantizer-agnostic strategy that adds a fixed uniform bias to activations, thereby reducing the quantization error of heavy-tailed distributions.
- **FIMA-Q** (Wu et al., 2025) suggests optimizing the rounding function and scaling factor using effective Hessian-guided quantization loss.
- **ERQ** (Zhong et al., 2025) decouples activation and weight error reduction via Ridge-regression-based weight correction. It also integrates several effective PTQ techniques, including incorporating channel-wise scaling factors into the LayerNorm parameters to handle activation outliers, and applying group-wise uniform quantization with separate quantizers for weight outliers, along with rounding function optimization.

All baselines use per-tensor dynamic quantization with 8-/6-/4-bit integer precision. Following the original papers, we use 1,024 calibration samples for NoisyQuant and 32 calibration samples for RepQ-ViT, PTQ4ViT, and ERQ. For FIMA-Q, we use 1,024 samples for optimization and 128 samples for calibration.

Hardware. Most standalone vision encoder experiments are conducted on NVIDIA RTX 4090 GPUs, whereas VLM evaluations are performed on NVIDIA RTX A6000 GPUs. Due to its large memory requirements, the FIMA-Q baseline is evaluated on both NVIDIA RTX A6000 and NVIDIA A100 80GB GPUs.

G. High Cosine Similarity of Outlier Tokens

In this section, we provide a simple explanation for why outlier tokens maintain high cosine similarity across images, as shown in Table 1.

As a stylized example, consider two distinct outlier tokens, each modeled as a "spiked" vector in which a single element has high magnitude (i.e., C in Theorem G.1). Theorem G.1 says that the high cosine similarity over the outlier tokens can largely be attributed to two factors[4]: **(i) the presence of large-magnitude entries** and **(ii) the alignment of the corresponding entry indices (i.e., same or at least similar channels)**.

Lemma G.1. *Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and fix an index $i \in \{1, \ldots, d\}$. Moreover, let $\mathbf{1}_i \in \mathbb{R}^d$ denote the one-hot vector whose ith entry is 1 and all other entries are 0. For $C \in \mathbb{R}^+$, define $\mathbf{x}' = \mathbf{x} + C\mathbf{1}_i$, $\mathbf{y}' = \mathbf{y} + C\mathbf{1}_i$. Then the vectors asymptotically converge to each other, i.e.,*

$$\lim_{C \to \infty} \frac{\langle \mathbf{x}', \mathbf{y}' \rangle}{\|\mathbf{x}'\|_2 \, \|\mathbf{y}'\|_2} = 1. \tag{3}$$

Moreover, for $\mathbf{y}'' = \mathbf{y} + C\mathbf{1}_j$, where $j \neq i$, then the vectors become asymptotically orthogonal to each other, i.e.,

$$\lim_{C \to \infty} \frac{\langle \mathbf{x}', \mathbf{y}'' \rangle}{\|\mathbf{x}'\|_2 \, \|\mathbf{y}''\|_2} = 0. \tag{4}$$

Proof. For the same-index case (Equation (3)), we can write this explicitly as

$$\frac{\langle \mathbf{x}', \mathbf{y}' \rangle}{\|\mathbf{x}'\|_2 \, \|\mathbf{y}'\|_2} = \frac{\langle \mathbf{x}, \mathbf{y} \rangle + C(x_i + y_i) + C^2}{\sqrt{\|\mathbf{x}\|_2^2 + 2Cx_i + C^2} \sqrt{\|\mathbf{y}\|_2^2 + 2Cy_i + C^2}}, \tag{5}$$

where x_i (resp. y_i) denotes the ith entry of $\mathbf{x}$ (resp. $\mathbf{y}$). Dividing numerator and denominator by C^2, we get

$$\frac{1 + \dfrac{x_i + y_i}{C} + \dfrac{\langle \mathbf{x}, \mathbf{y} \rangle}{C^2}}{\sqrt{1 + \dfrac{2x_i}{C} + \dfrac{\|\mathbf{x}\|_2^2}{C^2}} \sqrt{1 + \dfrac{2y_i}{C} + \dfrac{\|\mathbf{y}\|_2^2}{C^2}}}. \tag{6}$$

As $C \to \infty$, all terms of order $1/C$ and $1/C^2$ vanish, and we get what we want. The case of the different-index (Equation (4)) can be handled similarly. □

Indeed, we find that across different images, outlier tokens tend to have similar magnitudes in certain coordinates (i.e., along specific channels shared across different tokens).

[4]There are some cases where each component has a small magnitude, yet the vectors are still similar to one another. In high dimensions, however, this rarely happens; for example, random vectors become almost orthogonal as the dimension tends to infinity. In vision encoders, the number of channels per token is typically large (e.g., 768 channels in CLIP and SigLIP families), which results in low similarity among non-outlier tokens (e.g., "Normal tokens" in Table 1).

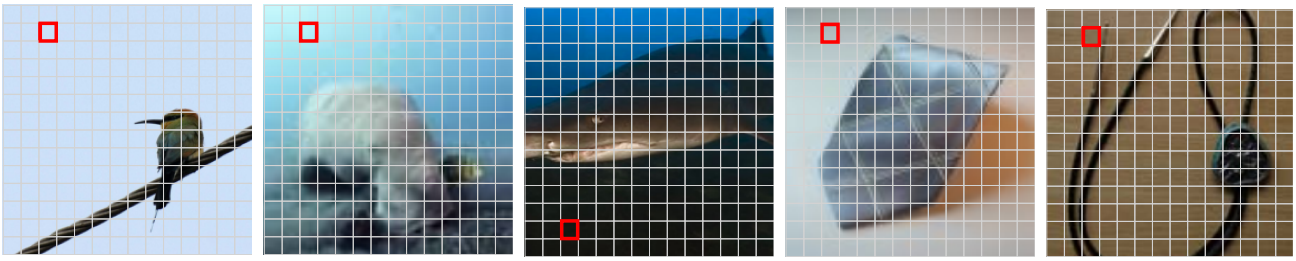

Figure 6. Visualization of curated prefix tokens from ImageNet-1k

H. Visualization of Curated Prefix Tokens

In Figure 6, we visualize the top 5 prefix tokens selected by the method described in Section 4.1, ranked by their effectiveness under quantization as measured by W8A8 zero-shot classification accuracy on ImageNet-1k. The results are consistent with prior findings (Darcet et al., 2024; Jiang et al., 2025), revealing that the register tokens are located in background regions. We find that the selected register tokens commonly correspond to low-frequency regions surrounded by semantically uninformative patches.

I. Segmentation Results

To analyze the impact of token deletion on dense prediction, we conduct semantic segmentation experiments on ADE20K[5] and report the mIoU results in Table 10a. We find that the sink token shows a noticeably lower mIoU score, consistent with prior observations that sink tokens can degrade segmentation performance (Jiang et al., 2025; Bolya et al., 2025). To remove the sink token while maintaining spatial structure, we replace its position with an interpolated token obtained by averaging the surrounding patch embeddings, and report the resulting performance in Table 10b. Our method effectively suppresses the outliers induced by sink-token deletion, thereby improving post-training quantization performance on dense prediction tasks.

Table 10. **Analysis of sink tokens and RegCache for segmentation.** Results on ADE20K using W8A8 quantized SigLIP-B/16.

(a) **mIoU by token**

Setting	mIoU
Sink	16.67
Random	12.02
Non-sink	30.15

(b) **Increased mIoU with RegCache**

Method	mIoU
FP32	32.85
Naïve	30.30
RegCache w/ interp.	**32.46**

[5]We use ADE20K-*MIT Scene Parsing Challenge 2016*, which is the subset of the ADE20K benchmark (Zhou et al., 2017), available at http://sceneparsing.csail.mit.edu/.

J. Impact of Activation Outliers on Quantization

To assess the impact of outlier on quantization, we control the quantization range at the token level. In Table 11, "w/ outliers" applies standard per-tensor quantization. In contrast, the "w/o outliers" setting excludes outlier tokens from the per-tensor range and quantizes them separately via per-token quantization. As a result, quantizing the outlier tokens separately shows a substantial performance improvement, which demonstrates that mitigating outliers is a critical problem in quantization.

Table 11. Comparison of quantization performance w/ and w/o outliers.

Model	Full Precision	w/ outliers	w/o outliers
CLIP-B/16	68.32	34.01	55.83 (+21.82)
OpenCLIP-B/16	70.22	46.12	65.43 (+19.31)
SigLIP-B/16	76.05	69.71	74.54 (+4.83)
SigLIP2-B/16	78.47	26.04	74.54 (+48.50)
DINOv2-B/14	83.26	19.20	76.58 (+57.38)

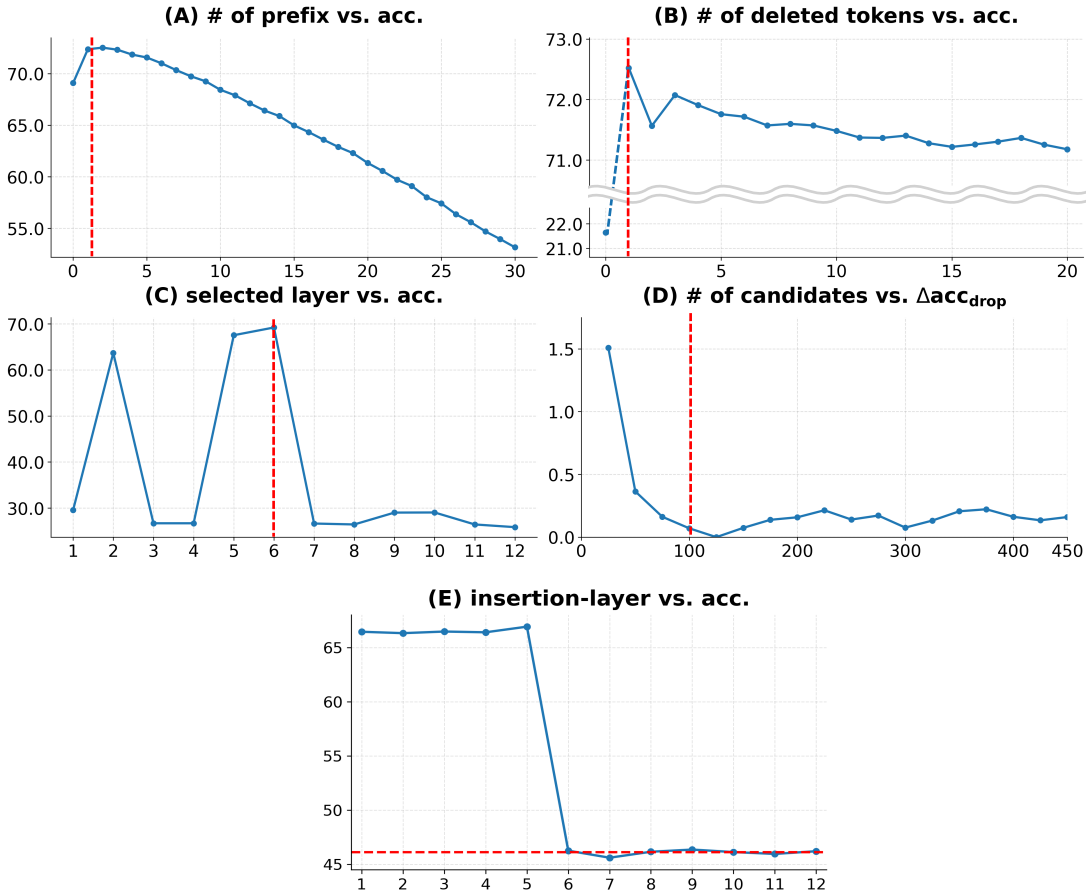

Figure 7. **Parameter-sensitivity analysis.** Results are evaluated by zero-shot accuracy on ImageNet-1k under W8A8 quantization.

K. Parameter-Sensitivity Analysis

In Figure 7, we present the parameter analysis described above. Figure 7 (A) and (B) show that RegCache is robust to the number of prefix tokens and the number of deleted tokens, respectively. Each curve exhibits a clear sweet spot: performance first improves as our proposed components become effective, but then decreases when excessive prefix tokens or token deletion introduces noise or information loss. Additionally, Figure 7 (C) shows that the quantization-sensitive layer must be carefully selected through search, as performance drops substantially when other layers are chosen. Furthermore, Figure 7 (D) shows that our selected number of candidates, i.e., 100 tokens, corresponds to the minimal point at which performance begins to plateau near its maximum. For Figure 7 (E), we visualize how accuracy changes depending on the layer at which prefix insertion begins. Accuracy drops sharply when insertion starts at layer 6, which is immediately after the quantization-sensitive layer. This indicates that prefix tokens are effective only when they are inserted before the sensitive layer. Based on this observation, we start prefix insertion from an intermediate layer before the sensitive layer, thereby reducing computational cost while preserving its effectiveness.

L. RegCache Selected Based on Reconstruction Loss

To further demonstrate the extensibility of our approach, we additionally evaluate a quantization reconstruction loss–based search as an alternative selection strategy. Specifically, we search for the number of prefixes and deleted tokens that minimize the MSE between output features produced by FP32 processing and those produced under a quantized setting. This criterion depends only on internal activations and does not require access to target-domain labels or downstream metrics, enabling a lightweight and broadly applicable procedure.

The results in Table 12 show that our method remains effective under this label-free selection rule, supporting its practical deployment in scenarios where validation data is unavailable or restricted. Moreover, even without ImageNet-1k-based tuning, our method improves accuracy on average across quantization settings.

Table 12. **Results of reconstruction loss–based search.** Zero-shot classification accuracy on ImageNet-1k for various vision encoders. The best results are marked in **bold**. Best/Average Δ denote the gaps between the best/average performance of each baseline with and without RegCache, excluding the Naïve cases. Here, RegCache rc denotes the results with RegCache selected based on reconstruction loss.

	CLIP-B/16			SigLIP-B/16		
Method	**W4A4**	**W6A6**	**W8A8**	**W4A4**	**W6A6**	**W8A8**
FP32		68.32			76.05	
Naïve	0.09	0.17	34.01	0.13	0.77	69.71
w/ RegCache rc	0.09	0.27	59.47	0.13	23.29	74.31
PTQ4ViT	0.37	51.60	67.69	0.19	68.68	75.57
w/ RegCache rc	0.32	58.33	67.77	2.19	72.27	75.93
RepQ-ViT	1.83	53.25	67.39	21.36	73.32	75.23
w/ RegCache rc	15.09	66.81	68.00	30.91	74.59	75.99
NoisyQuant	0.34	46.19	63.20	1.78	71.10	75.50
w/ RegCache rc	1.29	57.19	65.80	8.61	72.18	75.69
FIMA-Q	50.41	66.51	67.63	76.05	76.06	76.06
w/ RegCache rc	**62.27**	**66.90**	67.78	**76.12**	**76.12**	**76.12**
ERQ	1.56	39.64	67.99	57.76	74.44	75.95
w/ RegCache rc	47.11	65.90	**68.03**	60.06	75.24	76.04
Best Δ	+45.55	+26.26	+2.18	+9.55	+3.59	+0.76
Average Δ	+14.32	+11.59	+0.61	+4.15	+1.36	+0.29

M. Using FP16 Instead of Token Deletion

As a flexible variant of our method, we replace token deletion with FP16 token computation and report the results in Table 13. This option yields a modest accuracy improvement, but introduces a bottleneck in computing the FP16 tokens to propagate, thereby slowing down the overall process.

Table 13. **Performance of FP16 mixed precision instead of token deletion.** Accuracy of zero-shot image classification on ImageNet-1k under W8A8 quantization.

Method	**SigLIP**	**SigLIP2**
FP32	76.05	78.47
Naïve	69.71	26.04
w/ RegCache	74.38	72.35
w/ RegCache (FP16 MP)	74.51	72.82

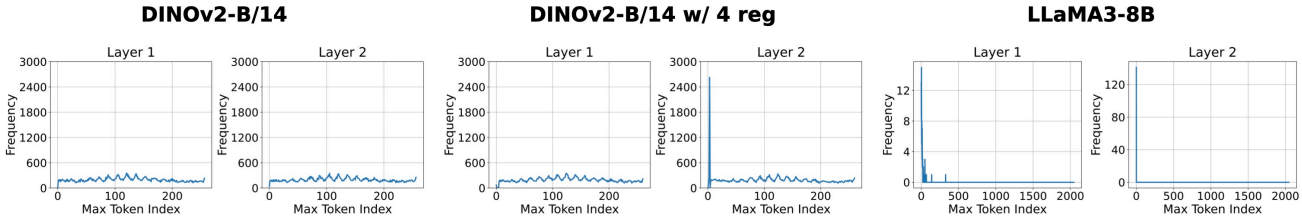

Figure 8. **The frequency of top-1 max tokens in the input tensor of FC2 layers in different models**. We evaluated DINOv2 on ImageNet-1k and LLaMA3-8B on WikiText-2 dataset.

N. Outliers in Vision Encoders vs. LLMs: A Tokenization Perspective

As shown in Figure 2, various vision encoders exhibit outliers in the intermediate layers. Revisiting recent studies about outlier tendency in LLMs (Sun et al., 2024; Gu et al., 2025), it is natural to ask: why do outliers consistently emerge in the intermediate layers of vision encoders, rather than in the early layers as in LLMs? In this section, we investigate this phenomenon by reconsidering the difference in *tokenization strategies* between vision encoders and LLMs.

Roughly speaking, LLMs map input sequences to tokens drawn from a discrete, fixed vocabulary. In contrast, vision encoders and other ViT-based models process inputs by mapping them to continuous token embeddings using a (convolutional) neural network. We hypothesize that differences in the emergence of sink tokens can be attributed to their fundamentally distinct tokenization processes.

To test this hypothesis, we compare the outlier behavior of DINOv2, pretrained both with and without learned register tokens, and LLaMA3-8B (Dubey et al., 2024). In this setup, the register tokens act as "fixed outlier sinks" (Darcet et al., 2024), effectively forming a closed-set vocabulary for outlier attraction, analogous to the tokenization setup in LLMs. As shown in Figure 8, when ViTs are equipped with four learned register tokens, they begin to exhibit outliers in early layers (i.e., the second layer), mirroring the behavior observed in LLMs. This supports our hypothesis that continuous tokenization in ViTs plays a crucial role in the emergence of outliers in the intermediate layers.

O. Weight-only Quantization

We demonstrate the effectiveness of RegCache when combined with a weight-centric method, i.e., weight-only quantization, which is commonly used in LLMs to reduce memory usage and deployment cost. Specifically, we adopt AWQ (Lin et al., 2024), a widely used weight-only quantization method, as the baseline, using a group size of 128 and bitwidths of 8, 6, 4, and 3. Across all configurations, RegCache consistently improves performance over vanilla AWQ, demonstrating its complementary benefits even in memory-constrained quantization settings. However, as noted in Section 1, unlike autoregressive LLMs, vision encoders are typically compute-bound, making weight-only quantization less effective.

Table 14. Zero-shot image classification accuracy (%) under weight-only quantization (AWQ).

Model	**FP32**	**Method**	**Weight-only (AWQ) Bits**			
			W3A16	**W4A16**	**W6A16**	**W8A16**
CLIP-B/16	68.32	AWQ	62.08	66.73	67.80	68.05
		+ RegCache	63.08 (+1.00)	67.06 (+0.33)	67.94 (+0.14)	68.14 (+0.09)

P. With Hadamard Rotation Quantization

Motivated by recent works in the LLM literature on utilizing Hadamard rotations for outlier mitigation (Ashkboos et al., 2024; Liu et al., 2025), we have explored whether RegCache can also be combined with such methods. In particular, we follow the adaptation of ViDiT-Q (Zhao et al., 2025) to apply Hadamard rotations to all linear layers in the transformer blocks.[6] As reported in Table 15, RegCache improves accuracy in most settings when used as a plug-in method, indicating that suppressing extreme outliers in the rotated domain can further enhance quantization performance.

Table 15. Zero-shot image classification accuracy (%) on top of QuaRot (Ashkboos et al., 2024)

	CLIP		OpenCLIP	
Method	**W6A6**	**W8A8**	**W6A6**	**W8A8**
FP32	68.32		70.22	
Naïve	0.17	34.01	0.47	46.12
QuaRot	55.86	66.38	59.49	**68.50**
QuaRot + RegCache	**56.05**	**66.84**	**59.76**	**68.50**

[6]There is one exception: we do not use the Walsh–Hadamard matrix for the value and output projection matrices, as in the ViDiT-Q implementation. Also, we have empirically observed that naïvely applying QuaRot to vision encoders tends to severely damage the model performance.

Q. Combining with LLM Outlier-mitigation Methods

We evaluate whether SmoothQuant (SQ)—a widely used outlier-mitigation PTQ technique for LLMs—transfers to ViT-based vision encoders. As shown in Table 16, SQ alone yields only marginal improvements over the naïve quantization baseline, consistent with prior observations on ViTs (Tai et al., 2024). While RegCache provides clear additional gains when applied on top of SQ, we further note that ViT-specific outlier mitigation methods such as RepQ-ViT (RepQ) may be a more effective direction for vision encoders.

Table 16. **W8A8 quantization on vision encoders.** Zero-shot ImageNet-1k accuracy (%) under W8A8 activation quantization.

			LLM PTQ method		ViT PTQ method	
	FP32	**Naïve**	**SQ**	**SQ + ours**	**RepQ**	**RepQ + ours**
CLIP	68.32	34.01	35.54	60.30 (+24.76)	67.39	68.10 (+0.71)
OpenCLIP	70.22	46.12	46.50	67.07 (+20.57)	68.70	70.06 (+1.36)
SigLIP	76.05	69.71	70.55	74.65 (+4.10)	75.23	75.94 (+0.71)
SigLIP2	78.47	26.04	28.20	72.45 (+44.25)	76.43	77.13 (+0.70)

R. Comparison with Test-Time Registers

Our experimental results demonstrate that RegCache consistently outperforms the test-time register (Jiang et al., 2025) variant in quantization settings. As discussed in Section 2, Jiang et al. (Jiang et al., 2025) propose register tokens as a concurrent approach to mitigating activation outliers. In an extended experiment reported in Appendix 2 (Ver. 5 (Jiang et al., 2025)), they further inject test-time registers as a form of KV cache while zeroing out outlier neurons, showing that high-norm residual-stream outliers can be suppressed. To assess the practical utility of this mechanism in quantized models, we evaluate it under identical conditions and compare it directly with RegCache.

The results show that RegCache remains consistently superior, suggesting that identifying the optimal curation tailored to a quantization-based setup is more effective than relying on a test-time register based outlier suppression mechanism.

Table 17. **Comparison with test-time register.** Accuracy of zero-shot image classification on ImageNet-1k under W8A8 quantization.

Method	CLIP-B/16	SigLIP-B/16
FP32	68.32	76.05
Naïve	34.01	69.71
Test-time register	44.30	73.81
RegCache	**59.71**	**74.38**