

POCA: Post-training Quantization with Temporal Alignment for Codec Avatars

Jian Meng^{1,2}, Yuecheng Li¹, Chenghui (Leo) Li¹, Syed Shakib Sarwar¹, Dilin Wang¹, and Jae-sun Seo²

¹ Meta Codec Avatars Lab, Pittsburgh, USA

² Cornell Tech, New York City, USA

Abstract. Real-time decoding generates high-quality assets for rendering photorealistic Codec Avatars for immersive social telepresence with AR/VR. However, high-quality avatar decoding incurs expensive computation and memory consumption, which necessitates the design of a decoder compression algorithm (e.g., quantization). Although quantization has been widely studied, the quantization of avatar decoders is an urgent yet under-explored need. Furthermore, the requirement of fast “User-Avatar” deployment prioritizes the post-training quantization (PTQ) over the time-consuming quantization-aware training (QAT). As the **first** work in this area, we reveal the sensitivity of the avatar decoding quality under low precision. In particular, the state-of-the-art (SoTA) QAT and PTQ algorithms introduce massive amounts of temporal noise to the rendered avatars, even with the well-established 8-bit precision. To resolve these issues, a novel PTQ algorithm is proposed for quantizing the avatar decoder with low-precision weights and activation (8-bit and 6-bit), without introducing temporal noise to the rendered avatar. Furthermore, the proposed method only needs **10% of the activations** of each layer to calibrate quantization parameters without any distribution manipulations or extensive boundary search. The proposed method is evaluated on various face avatars with different facial characteristics. The proposed method compresses the decoder model by $5.3\times$ while recovering the quality on par with the full precision baseline. In addition to the avatar rendering tasks, POCA is also applicable to image resolution enhancement tasks, achieving new SoTA image quality. <https://mengjian0502.github.io/poca.github.io/>

1 Introduction

Photorealistic telepresence [14, 15, 23, 27] has emerged as the cornerstone technology in augmented or virtual reality (AR/VR), enabling virtual interactions across different devices. Specifically, the ultimate goal of photorealistic rendering is animating the high-fidelity Codec Avatars with multiple partial facial images captured by the head-mounted cameras (HMC) [29]. The variational autoencoder-based model [14, 15] has been proved and deployed to VR device (e.g., Quest Pro) as a feasible solution. During inference, the encoder model projects the HMC-captured facial motion into the latent space, and the decoder

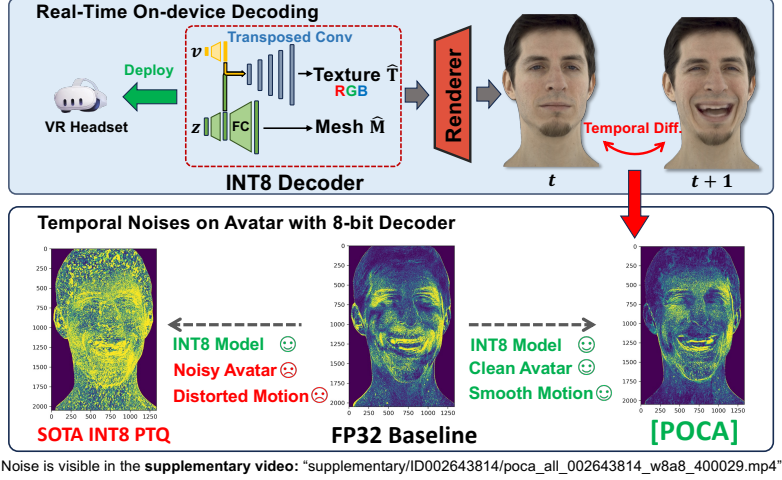


Fig. 1: Decoding and rendering with low-precision avatar decoder. The SoTA 8-bit PTQ algorithm distorts the avatar with massive and random temporal noise, while the proposed method compresses the decoder with the high-quality avatar without noise.

reconstructs the latent code into view dependent high-quality texture and mesh, which are then rendered into the 3-D Codec Avatar on screen. In practice, the critical ingredient of photorealistic telepresence is transmitting the encoded latent information from one VR to another and rendering the avatar in real-time. In practice, the VAE-based Codec Avatar model has been employed for generative Pixel Codec Avatar [15] or 3D Gaussian-based Avatar [22]. Naturally, the quality and latency of telepresence **mainly depends on the efficiency of the decoder**, which impacts the real-time visual quality.

Despite the success of the deep appearance model [14], the high computational cost and memory consumption of avatar decoding hinder the fast on-device telepresence. **Naïvely deploying the full-precision decoder to the headset leads to a low frame rate, causing degraded user experience.** On the other hand, relying on cloud-based computation adds additional “headset-cloud” communication latency and increases the risk of avatar data breaching.

The conflict between high-fidelity rendering and the urgent needs of on-device computing necessitates the high-performance compression algorithm designed for the avatar decoder. In particular, quantization [2, 18, 28, 30] compresses the model with a unified data compression ratio on weights and activation, leading to easy hardware deployment [24]. Unfortunately, the SoTA quantization algorithms mainly focus on classification [11, 18], object detection [26], and natural language processing tasks [30]. Recent works have investigated high-resolution static image rendering with the low-precision model [25]. **However, the high-quality quantization algorithm for avatar decoding remains unexplored.** In fact, our research shows that the current efforts on quantization are **highly incompatible** to the Avatar decoder with the following challenges:

Massive temporal noises introduced by INT8 quantization. Although the popular 8-bit quantization has been widely investigated with quantization-aware training [20] and post-training quantization [2, 18, 24], **our discovery** shows that the well-established INT8 precision introduces a massive amount of **temporal noises** to the rendered avatar, even with the SoTA methods. As shown in Figure 1, the facial motion generated by the 8-bit decoder is filled with temporally flickering noises between two consecutive frames, which immerses the original facial expressions and degrades the quality of the rendered avatar. Most importantly, the temporal noise is **not visible** in the static single frame, distorting the avatar in a continuous manner. As the first work in this area, this paper reveals the **root cause** of the temporal avatar noise, as described in Section 3.

Difficulty of characterizing temporal noise. Common quantization noises in conventional classification and object detection tasks can be modeled by a standard statistical model [12]. However, the temporal avatar noises introduced by quantized decoder exhibit inconsistent intensities and granularities in different facial regions. As a result, it is almost impossible to model the temporal noises globally without additional recurrent information. Furthermore, applying an additional filtering strategy after quantization introduces additional energy and latency consumption to the AR/VR devices, which deteriorates the efficiency of Codec Avatar telecommunication. **Therefore, the noise suppression process should be achieved all at once during post training quantization.**

To resolve the aforementioned problems, we propose *POCA*, a novel Post Training Quantization (PTQ) algorithm designed for **high quality** and **temporally noise-free** Codec Avatar decoding with outstanding simplicity and efficiency. The major contributions of this work are listed as follows:

- As the **first** work that investigates the low-precision avatar decoding, we reveal the **root cause** of the temporal noises induced by the temporally **dynamic** and **long-tailed** distribution.
- *POCA* eliminates the temporal facial noise with 8-bit and 6-bit precision, outperforming the SoTA method with fully recovered avatar quality [16].
- *POCA* only requires $\leq 10\%$ of layer-wise feature map data and $\leq 5\%$ of the total training data for PTQ.
- *POCA* is evaluated on a variety of avatar identities with a full range of viewing directions and dynamic expressions, exhibiting a high degree of robustness and consistent noise-free avatar animation across different genders, age groups, hairstyles, skin colors, and view directions.
- *POCA* is also applicable to image super resolution task [8, 25], achieving the SoTA image quality with **80%** calibration sparsity.

2 Related Work

2.1 Codec Avatar

Photorealistic human face modeling and animation are challenging computer vision and computer graphics tasks in the sense of achieving minimum visual

distortion for reconstructed 3-D human faces. Early works mainly focus on accurate 3-D reconstruction with blend shape models [9], which are computationally expensive and impossible for mobile on-device telepresence. Empowered by the deep learning models, recent research [14, 15] considers human face reconstruction as data-driven generative task with the variational auto-encoder neural network. In particular, the variational encoder [6] is trained for encoding the multi-view HMC images into the latent information $\mathbf{z}$, which describes the state of the face with rich expression and motion. Subsequently, the decoder network reconstructs the 3-D faces based on the latent code and the given view direction $\mathbf{v}$, where the assets for rendering 3D face avatar are decoded separately by the dedicated *mesh decoder* and *texture decoder*.

The compact latent code is transmitted between VR devices with low communication bandwidth requirements. However, the desire for real-time, on-device 3-D face avatar animation is hindered by the expensive computation cost and high memory consumption of the decoder. Various efforts have been established to improve the computation efficiency of the VAE model [5, 14]. Pixel Codec Avatar [15] proposes a novel architecture to allow lightweight decoder by creatively leveraging rendering pipeline features, skipping computations for invisible pixels while maintaining the high-fidelity for visible regions. Recent research optimizes the encoder architecture via neural architecture search (NAS) [3], leading to reduced computational costs. Different from such prior works, our work proposes a decoder-dedicated quantization algorithm that enables temporal alignments and noise-free avatar decoding with low-precision model, which further enhances instant and high quality on-device deployment of Codec Avatars for resource-constrained mobile AR/VR headsets.

2.2 Quantization

Quantization has been widely investigated together with the evolution of deep learning. Starting from the well-established 8-bit precision [4, 24] all the way to the aggressive binary neural network [32], the quantized model largely improves the computation and memory efficiency, making the complex DNN models deployable to the resource-constrained edge devices (e.g., AR/VR headset). The core idea of quantization is constructing the accurate matching between the high-precision data (e.g., FP32) and the low-precision representation. Normally, the quantization process can be generalized into the “quantize-and-dequantize” workflow connected by the shared scaling factor in between. Mathematically, given the full precision weights $\mathcal{W}^l$ and input activation $\mathcal{X}^l$ of a layer l , we have:

$$\mathcal{W}_Q^l = \left\lfloor \frac{\mathcal{W}^l}{S_w^l} \right\rfloor \in \text{INT}_{\mathbf{n}}, \quad \mathcal{X}_Q^l = \left\lfloor \frac{\mathcal{X}^l}{S_x^l} \right\rfloor \in \text{INT}_{\mathbf{n}} \quad (1)$$

Where S is the scaling factor determined by data precision $\mathbf{n}$ and the numerical range. Specifically, S characterizes the “interval” between adjacent levels in the floating point dequantized values, bounded by the range $[\text{lb}, \text{ub}]$:

$$S = |\text{ub} - \text{lb}| / 2^{\mathbf{n}} - 1 \quad (2)$$

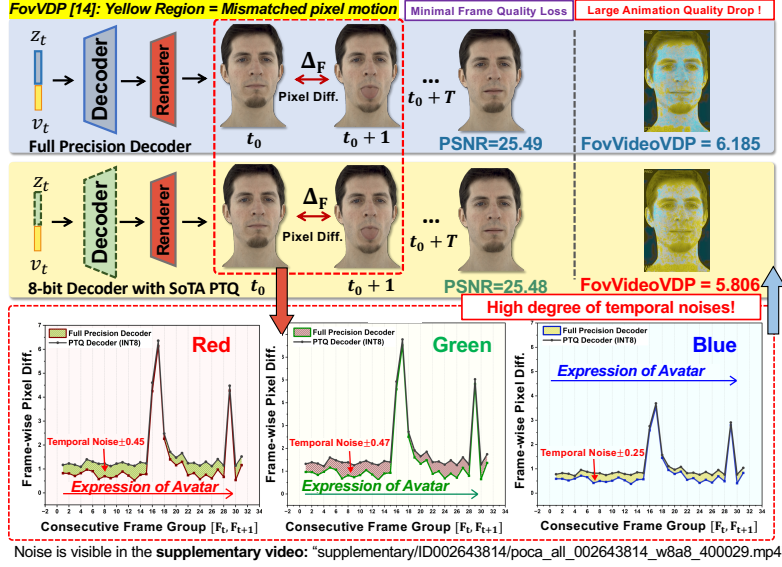


Fig. 2: Top: The rendered avatar exhibits minimal quality (PSNR) reduction in static frames and large distortion with animation. **Bottom:** SoTA 8-bit quantization [2, 18] amplifies the pixel shifting between frames in the rendered avatar across RGB channels.

Since the number of levels is directly determined by the precision n , the **essence** of quantization accurately approximates the floating-point distribution with the dequantized tensors while minimizing the model-level error. To that end, a variety of quantization algorithms have been proposed to locate the optimal scaling factors with QAT [13] or PTQ [11, 18, 28], including boundary learning [1, 10], scaling factor learning [2], and quantization error modeling [12], etc. Recent efforts introduce additional scaling to avoid the error caused by the extreme distribution [30]. Promoted by the fast compression process, PTQ exhibits high compatibility for quantizing the avatar VAE model compared to QAT. Ideally, compressing the Codec Avatar models with PTQ would lead to instant “user-avatar” conversion for AR/VR devices, compared to time-consuming QAT. Recent VAE-based quantization focuses on the video compression task with quantized latent code [7, 21], while the quantization of the codec-avatar decoder model has not been explored despite the urgent needs on edge device and high rendering quality.

3 Root Cause of Temporal Avatar Noise

3.1 Temporal Noise is Dynamic Pixel Color Oscillation

Recent 3D rendering tasks (e.g., NeRF) [17] uses peak signal-to-noise ratio (PSNR) to evaluate the quality between the rendered output and the ground truth im-

age. However, the temporal distortion of Codec Avatar **cannot** be effectively quantified with the same metrics.

With the low-precision decoding, the temporal noise shows minimal impact on the static frame quality, but largely **distorts the dynamic motion** of the avatar, as shown in Figure 2 and supplementary material. Prior works **only** focus on the static image quality [25] with low-precision models, but the dynamic quality reduction has not been investigated well.

In addition to the FovVideoVDP score [16], the temporal pixel oscillation can be characterized by the mean pixel value differences between two consecutive frames F_t and F_{t+1} of the rendered avatar:

$$\Delta_F^c = \frac{1}{\mathcal{H}\mathcal{W}} \times |F_{t+1} - F_t| \quad (3)$$

Where $\mathcal{H}$ and $\mathcal{W}$ represent the height and width of the rendered frame. We evaluate the pixel value differences separately for RGB channels such that $c \in \{R, G, B\}$. The trend of the pixel changes characterizes the basic transition of the facial expression (e.g., Tongue in and out), as shown in Figure 2 (**Bottom**).

With the 8-bit VAE decoder, the frame-wise pixel dynamics exhibit a similar trend as the full-precision decoder baseline. However, the quantized decoder adds an additional “offset” to the transition of the facial motion. In other words, the additional distortion is introduced to the avatar on top of the natural motion, which leads to random texture fluctuation in the rendered Codec Avatar. Figure 2 (**Bottom**) shows that the temporal noises uniformly exist across the entire facial motion with low intensity of ± 0.44 , ± 0.47 , and ± 0.25 average pixel shifting on the R, G, B channels, respectively. We are fully aware of the existence of other motion capture matrices (e.g., optical flow), but the temporal noises are **significantly fine-grained**, which requires detailed pixel-level analysis.

Formally, such random pixel flickering can be captured by the FovVideoVDP [16] score between the rendered avatar and ground truth image. As shown in Figure 2, the INT8 avatar decoder largely degrades the animation quality of the avatar, even with the SoTA 8-bit quantization method [2, 18]. The high density of the yellow region indicates the strongly mismatched animation caused by the temporal noise. The continuous pixel oscillation and temporal noises worsen the fidelity of the rendered Codec Avatar in motion.

3.2 Root Cause of Noisy Avatars

Since the temporal noises are induced by the pixel oscillation of the avatar in motion, an intuitive understanding of the noises is the “unstable rounding” during quantization [12, 19]. **However, the rounding instability is still the phenomenon of quantization instead of the root cause of the noise.** In this work, we unravel the **fundamental reason** of the temporal noise induced by low-precision avatar decoding.

Dynamically long-tailed distribution of human face. Human facial features naturally display a diverse range of distributions characterized by dif-

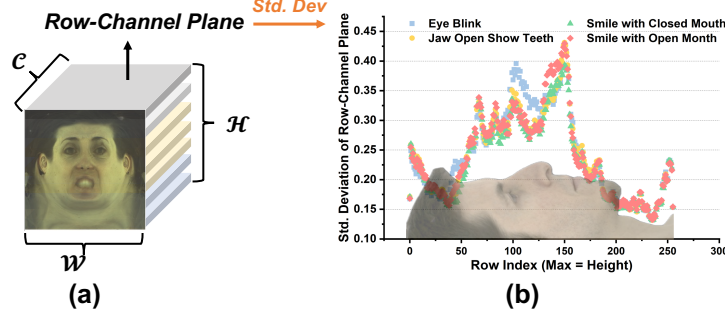


Fig. 3: (a) Row-channel plane of the texture decoder activation and (b) Regional standard deviation of the row-channel planes with respect to different facial expressions.

ferent facial regions (for instance, hair compared to cheeks). The broad spectrum of variability in human facial characteristics formulates various statistical properties. For the avatar decoding, the texture decoder of the VAE model [14] decodes face avatar texture from the latent space via transposed convolutional layers, which consecutively up-scales the activation feature map from 4×4 to 1024×1024 (height and width). Therefore, the statistical variation of the facial texture is also embedded into the activation of the earlier layer.

Proof of facial dynamics. We prove the dynamic range variations of the texture by analyzing the regional distribution of the latent layer activation. Assume $\mathcal{C}$, $\mathcal{H}$, and $\mathcal{W}$ represent the channel depth, height, and width of the activation tensor. We analyze the statistics by decomposing the entire face texture feature map into multiple “row-channel planes”. Each plane contains the regional statistics from hair to jaw, with the total number of planes $= \mathcal{H}$.

Figure 3(a) shows the row-channel decomposition based on the intermediate activation feature map of the texture decoder with the shape of $16 \times 256 \times 256$. On top of that, we compute the *standard deviation* of each plane, which characterizes the local tailedness and dynamic range in each region of the face, with respect to different facial expressions. As depicted in Figure 3(b), smiling and eye blinking lead to the long-tailed dynamics in the mouth and eye regions, respectively. As a result, the distribution of the entire face is **dynamically long-tailed**, which is harmful for quantization. In other words:

The distribution of a decoder feature map contains both constrained and long-tailed distribution, while the tailedness varies between different facial expressions.

In other words, the **root cause** of the temporal noises is the conflict between the long-tailed, highly dynamic distribution of **activations** and the global error minimization during PTQ. To minimize the global quantization error, the long-tailed distribution stretches the quantization interval (scaling factor) in order to cover larger range, which in turn, elevates the local rounding instability and leads to unstable numerical representation of the pixels. We prove this hypothesis in Section 4.3 with the proposed method.

4 Proposed Method: POCA

To resolve the temporal noises caused by low-precision avatar decoding, this section presents POCA, a novel post-training quantization algorithm that successfully compresses the deep appearance decoder and eliminates the temporal noises from the rendered Codec Avatar. In particular, POCA is empowered by two novel techniques: a) View-aware PTQ and b) PTQ with shape-wise filtering.

4.1 View-aware PTQ

As pointed out in Section 3, the face texture contains both region-dependent dispersion and facial expression-driven dynamics in the entire face. As a result, the sample space of the training data and layer-wise activation contains high variation with redundant information. In particular, the texture decoder is trained by the **whole face** texture map with the size of 1024×1024 . However, the final 3D avatar is rendered based on the visibility information from the given view direction. Therefore, quantizing the decoder without considering the visibility for inner layers leads to the incorrectly learned quantization parameter, which further generates the mismatched quantization decision between PTQ and actual rendering. Motivated by that, we propose the **view-aware PTQ** for the avatar *texture decoder*.

Basic Setup. Given the pre-trained full-precision VAE decoder (texture and mesh), we first randomly select m training samples from the full-sized dataset with size $\mathcal{D}$ ($m \ll \mathcal{D}$). Each training sample contains the input and pre-computed output of the decoder ($[z_t, v_t], [\hat{\mathbf{T}}_t, \hat{\mathbf{M}}_t]$), where z_t , v_t , $\hat{\mathbf{T}}_t$, and $\hat{\mathbf{M}}_t$ represents the encoded latent code, view direction, decoded texture, and decoded mesh for sample t , respectively. Following the standard PTQ strategy, we extract input $\mathcal{X}^l$ and output activation $\mathcal{Y}^l$ of each layer as the calibration data. We adopted AdaRound [18] and LSQ [2] as the weight and activation quantizers, respectively. The learnable quantization parameters (α , $\mathcal{S}^l$) are optimized by minimizing the mean square error (MSE) between the quantized layer output $\hat{\mathcal{Y}}^l$ and full precision output $\mathcal{Y}^l$:

$$\arg \min_{\alpha, \mathcal{S}^l} \frac{1}{\mathcal{HWC}} \sum (\mathcal{Y}^l - \hat{\mathcal{Y}}^l)^2 \quad (4)$$

View-dependent visibility mask. The computation of visibility is inherited from the rendering pipeline utilizing the pre-trained VAE model. Specifically, the decoded mesh vertices $\hat{\mathbf{M}}_t$ are rasterized to formulate the barycentric coordinates, which are leveraged to interpolate the UV coordinates Φ across the entire avatar. Following this, the visibility is determined by sampling the view-specific pixels from the 2-D texture map based on the interpolated UV indexes, $\Phi = [\phi_x, \phi_y]$ ($0 \leq \phi_x, \phi_y \leq 1$).

The value of ϕ_x and ϕ_y represents the *normalized* coordinates of the 2-D texture map with the shape of $\mathcal{H} \times \mathcal{W}$. Different from the default pipeline, the

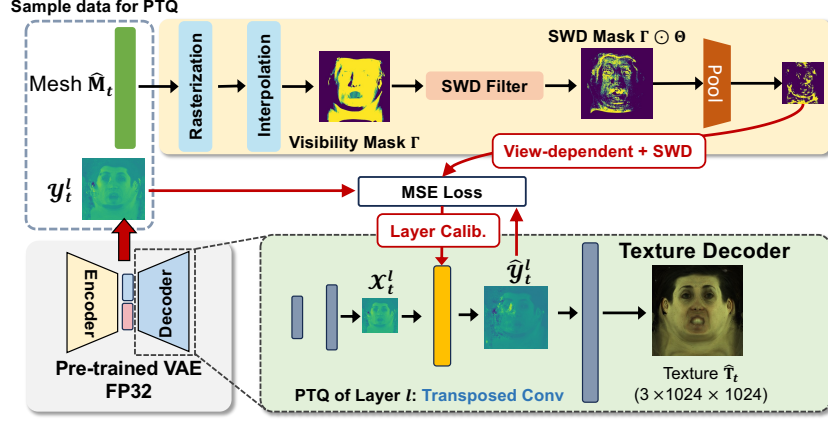


Fig. 4: Overall PTQ process of POCA based on layer-wise calibration (Layer Calib). POCA fully quantizes all the layers in the VAE decoder, where the proposed view-dependent + SWD filter is applied to the decoder with transpose convolution layers.

proposed POCA algorithm generates the binary “**visibility mask**” Γ to select the pixels in the 2-D texture map $\hat{T}_t$. Formally, we have:

$$\Phi = \text{Interpolate}(\text{Rasterize}(\hat{M}_t, \dots), uv) \quad (5)$$

$$\Gamma = \mathbb{1} \cdot \{(\gamma_x, \gamma_y) | \gamma_x \in \{\phi_x \times \mathcal{H}\}, \gamma_y \in \{\phi_y \times \mathcal{W}\}\} \quad (6)$$

Where $\mathbb{1}$ represents the indicator function that signifies whether the pixel at (γ_x, γ_y) should be active at the given visibility. $\{\phi_x \times \mathcal{H}\}$ and $\{\phi_y \times \mathcal{W}\}$ are the visibility coordinates (matrix indexes) scaled based on the interpolated UV coordinates Φ with the dimensionality of the texture map.

The visibility masks are generated during the data-fetching process. Since the model is already pre-trained, the visibility masks Γ are **accessible** with **each** full-precision texture $\hat{T}_t$ and triangle index uv at the output layer of the texture decoder. The final layer Γ (1024×1024) will be max-pooled to formulate the visibility mask for the earlier layer. For each layer of the texture decoder, POCA performs **PTQ with the visible pixel only**:

$$\arg \min_{\alpha^l, S^l} \frac{1}{\text{HWC}} \sum (\mathcal{Y}_T^l \odot \Gamma^l - \hat{\mathcal{Y}}_T^l \odot \Gamma^l)^2 \quad (7)$$

Where $\odot$ represents the element-wise product between tensors. Subsequently, the learnable quantization parameters α^l and S^l are trained based on the gradient that resulted from the view-dependent visibility Γ . As a result, POCA optimizes the quantization decision based on the **most relevant information** of the training sample, instead of being diversified by the widely dispersed texture space with long-tailed and dynamic distribution. Figure 4 depicts the proposed view-dependent PTQ. Although the quantizer is calibrated based on the visible pixels, **POCA only runs PTQ once**, and the resultant model will be deployed and rendered for **all views**.

4.2 PTQ with Shape-wise Dynamic Filtering

Besides the PTQ with the view-dependent visibility masking, we further suppress the impact of the dynamically long-tailed distribution by eliminating the pixels with a high channel-wise dynamic range, or “Shape-Wise Dynamics” (SWD). Given the full-precision output $\mathcal{Y}_T^l$ with the shape of $\mathcal{C} \times \mathcal{H} \times \mathcal{W}$, the SWD is characterized by the standard deviation score of each $\mathcal{C} \times 1 \times 1$ vector, which formulates the overall texture SWD map with the shape of $\mathcal{H} \times \mathcal{W}$.

Subsequently, POCA generates the SWD mask Θ by filtering out the pixels in the SWD map with the highest tailedness. The sparsity of the SWD mask is controlled by a tunable hyperparameter τ . E.g., $\tau = 0.7$ means masking out the top 70% pixels with the highest tailedness (standard deviation). The resultant SWD mask Θ will be multiplied with the view-dependent mask Γ and generate the final learning objective of the layer-wise post-training quantization of POCA:

$$\arg \min_{\alpha^l, S^l} \frac{1}{\mathcal{HWC}} \sum (\mathcal{Y}_T^l \odot \Gamma^l \odot \Theta^l - \hat{\mathcal{Y}}_T^l \odot \Gamma^l \odot \Theta^l)^2 \quad (8)$$

As shown in Eq. (8), POCA optimizes the quantizers by learning the most relevant information (view-dependent) and least diversified pixels (SWD masking) of the avatar, which suppresses the drawbacks of the long-tailedness and high dynamics of the avatar.

The masking strategy of POCA introduces a high degree of sparsity in layer-wise PTQ, controlled by visibility and the SWD threshold τ . Our experimental results in Section 5 show that the optimal performance is achieved with $\tau = 0.9$. As a result, POCA calibrates the low-precision layer based on highly sparse data $[\mathcal{X}^l, \mathcal{Y}_T^l]$ with $\geq 90\%$ sparsity. In other words, **POCA only requires $\leq 10\%$ of the activation for layer-wise PTQ**, showing low memory overhead for the efficient PTQ calibration process.

4.3 Rationale behind POCA

POCA alleviates the impact of rounding instability. With the proposed view-aware PTQ and SWD masking, POCA aims to optimize the layer-wise quantizers with the most relevant and quantization-friendly features. The POCA-trained activation quantizer (LSQ) has a generally smaller scaling factor S^l compared to the PTQ without POCA, as shown in Figure 5. The reduced scaling factors formulate the low-precision grid with small intervals (scaling factor) between adjacent levels, alleviating the numerical differences between adjacent levels in the dequantized high-precision domain. As a result, the impact of the aforementioned rounding instability is minimized and the temporal noises become invisible, which proves the aforementioned hypothesis in Section 3.2.

Sparse masking $\neq$ biased learning. Although two types of sparse masking are introduced into PTQ, the POCA-trained quantizers will not lead to a biased representation in the quantized model. As shown in Figure 4, the sparsified calibration mask ($\Gamma^l \odot \Theta^l$) contains sparse but uniformly sampled information

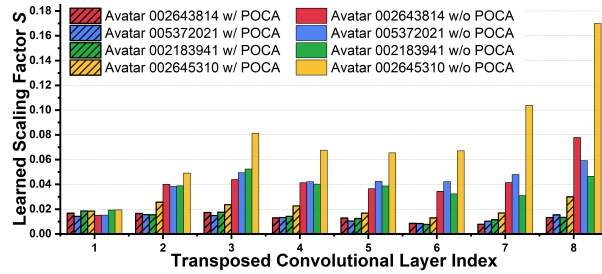


Fig. 5: Scaling factor S^l of the texture decoder trained by POCA.

across all the regions of the avatar texture. As a result, the learnable quantizers are uniformly trained with all the relevant facial information.

Highly-correlated data and high fidelity sparse learning. Unlike classification tasks, the training samples of the VAE model are highly correlated with respect to the single identity. As a result, the sparsified information of each sample can be compensated from other training data with different viewing perspectives, leading to sparse but robust PTQ with the quantized decoder.

5 Experimental Results

We evaluated the proposed POCA algorithm based on Multiface [29], a public avatar dataset captured by the multi-camera system. POCA is applied to eight avatars among three viewing directions with **all 65 facial** expressions of the dataset. We want to highlight that the eight avatars **comprehensively** cover different genders, age groups, hairstyles, skin types, and facial hairs.

For each identity, we use the corresponding pre-trained deep appearance VAE model as the starting point of POCA. Following the standard PTQ strategy, we quantize **all** layers of the decoders (texture and mesh) iteratively, where each layer is calibrated by 100 epochs with the sampled data. **POCA only acquires 512 training samples from each identity, which is $\leq 5\%$ samples of the dataset.** For each layer, the quantizers are optimized by Adam with $1e-4$ learning rate and eight samples per batch.

Prior Works for Comparison. As the **first** work in the low-precision avatar decoding, we select three prior quantization works that reported the SoTA INT8 performance with the classification tasks: (1) Adaptive Rounding (AdaRound) [18], (2) learned step size quantization (LSQ) [2], which are also incorporated for hardware deployment [24], and (3) W8A8 naïve QAT [20] in PyTorch. These selected prior works achieved superior INT8 performance [2, 18] on the ImageNet-1K dataset, even outperforming the full-precision baseline.

Evaluation Metric. We evaluate the proposed POCA algorithm based on the FovVideoVDP [16] between the rendered avatar and the ground truth video to quantify the temporal noise of the dynamic Codec Avatar. For each avatar identity, we compute the averaged FovVideoVDP score among all the 65 expressions in the Multiface dataset [29]. The performance of POCA is demonstrated with the rendered avatar videos in the supplementary material.

Table 1: FovVideoVDP score [16] between the rendered avatar and the ground truth based on eight avatars across all the expressions from three viewing perspectives. The proposed POCA algorithm achieves the highest quality compared to the SoTA PTQ and QAT methods with both 8-bit and 6-bit precision.

Avatar ID	Precision	2643814	2183941	5372021	2645310	8870559	6674443	2914589	6795937
		Male, Goatee	Black, Female	Mohawk, Male	Asian, Female	Older Woman	Male, Mid Hair	Blonde Girl	Male, Short Hair
		Front Top Side	Front Top Side	Front Top Side	Front Top Side	Front Top Side	Front Top Side	Front Top Side	Front Top Side
Baseline [20]	FP32	6.21 4.94 5.30	6.54 4.99 5.45	6.14 4.87 5.40	6.32 5.02 5.59	6.05 4.95 5.13	6.02 4.86 5.19	6.41 5.11 5.63	6.11 4.77 5.32
SoTA PTQ [2, 18]	W8A8	5.81 4.43 4.87	5.91 4.40 5.00	5.80 4.52 5.19	5.87 4.56 5.10	5.43 4.44 4.68	5.52 4.45 4.76	5.80 4.62 5.10	5.90 4.32 5.16
W8A8 QAT [20]		5.92 4.55 4.99	6.02 4.57 5.12	5.88 4.56 5.23	5.97 4.66 5.16	5.57 4.50 4.79	5.60 4.50 4.86	5.92 4.74 5.19	5.90 4.41 5.21
POCA		6.19 4.93 5.26	6.31 4.80 5.38	6.09 4.83 5.33	6.21 4.95 5.42	5.95 4.82 5.08	5.99 4.72 5.07	6.31 4.99 5.50	6.01 4.65 5.23
SoTA PTQ [2, 18]	W6A6	4.50 3.66 4.06	4.96 3.81 4.12	4.55 4.10 4.47	5.22 3.91 4.45	4.29 3.73 3.96	4.25 3.56 3.89	5.35 3.99 4.55	4.55 3.92 4.28
W6A6 QAT		5.08 3.75 4.39	5.54 4.01 4.23	4.94 4.19 4.49	5.36 4.07 4.66	4.36 3.85 4.09	4.46 3.61 4.08	5.46 4.10 4.64	4.74 4.07 4.33
POCA		6.06 4.55 5.02	6.29 4.42 5.13	6.01 4.61 5.16	6.02 4.66 5.21	5.45 4.49 4.99	5.66 4.43 4.93	6.19 4.80 5.26	5.92 4.41 5.11

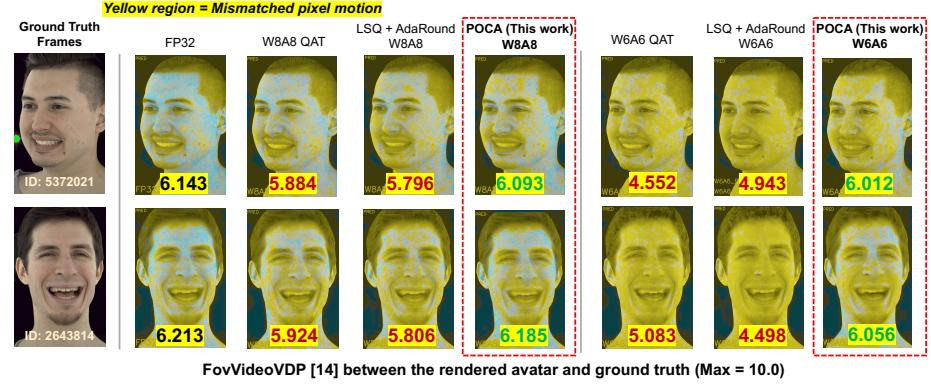


Fig. 6: Rendered avatars and temporal pixel differences between two consecutive frames of the avatar video. The lightened yellow region of POCA signifies the removed temporal noise as the full precision baseline.

5.1 Temporal Noise Reduction with High Fidelity

We evaluate the proposed POCA algorithm with eight avatars with different genders, hairstyles, and skin colors. Table 1 summarizes the FovVideoVDP score [16] of the proposed POCA algorithm with 8-bit and 6-bit avatar decoders. The SoTA INT8 PTQ quantization algorithm failed to preserve the avatar quality as the baseline full precision model, exhibiting the degraded FovVideoVDP score. Even with the expensive quantization-aware training (QAT), the quality of the rendered avatar cannot be restored.

Compared to the previous methods with 8-bit precision, POCA eliminates the temporal noises in the rendered avatar. POCA successfully compresses the decoder model while achieving on-par quality as the full precision decoder baseline. Figure 6 shows the high avatar quality and restored FovVideoVDP score (heat map) of the proposed POCA algorithm (closer to the FP32 baseline = better quantization). Regarding the efficiency of PTQ, **POCA only requires $\leq 10\%$ of the activations for layer-wise PTQ**, showing low memory overhead for the efficient PTQ calibration process.

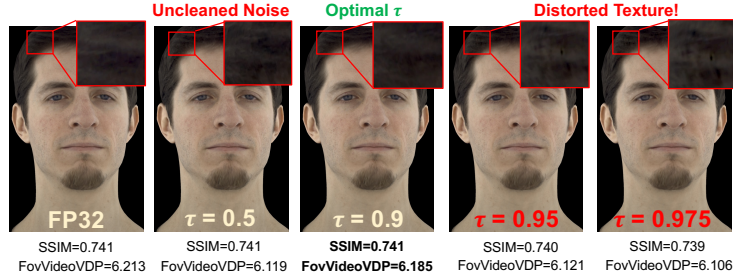


Fig. 7: The impact of the SWD threshold. Extreme filtering leads to distorted and missing texture in the resultant avatar.

Table 2: Training time comparison between POCA and QAT.

Method	Quantization Method	W/A	Time	GPU	Decoder Size (MB)
FP32 Baseline	-	32/32	-	-	32.78
Native W8A8 [20]	QAT	8/8	>10 hrs	2 × A6000	8.19 (↓ 4×)
POCA (This work)	PTQ	8/8	~13 min	1 × A6000	8.19 (↓ 4×)
POCA (This work)	PTQ	6/6	~13 min	1 × A6000	6.15 (↓ 5.3×)

POCA generates clean avatars **without** introducing any graphic filtering techniques [31]. From the perspective of quantization, POCA keeps the vanilla quantization strategy in Eq. (1) in both training and inference, compressing the model without introducing distribution manipulations [28] on the additional scaling [30] and bias [12]. Compared to traditional QAT, POCA quantizes the entire model in **only 13 minutes** (Table 2). The fast deployment workflow with the actual VR device (e.g., Quest Pro) can further accelerate the process of “user-to-meta verse”.

Future Applicability: POCA is **not** limited to the deep appearance model [14]. The proposed POCA algorithm is applicable to all mesh-based 3D reconstruction tasks with VAE models. For example, the proposed view-dependent visibility mask and shape-wise filtering can be easily transferred to the recent Pixel Codec Avatar (PiCA)³ [15], which has the similar autoencoder-based architecture.

5.2 Impact of the SWD Filtering Threshold

POCA introduced the tunable parameter τ in Eq. (8) to control the amount of activation pixels for layer-wise PTQ. Figure 7 demonstrates the impact of the SWD threshold τ with respect to the avatar quality (SSIM). POCA achieves the best tradeoff between filter intensity and avatar quality with $\tau^* = 0.9$. Quantizing the decoder with $\tau > \tau^*$ leads to missing texture pixels and distorted rendering with no quality improvement. The selected SWD threshold exhibits low sensitivity between different avatars and viewing perspectives, as demonstrated by the high-quality rendering in Figure 6.

³ PiCA results are available at <https://mengjian0502.github.io/poca.github.io/>

5.3 Applicability to Other Quality Enhancement Tasks

We extend the experiment by evaluating the proposed method on super-resolution tasks. The proposed “shape-wise-filtering” (SWD) exhibits strong performance in image quality enhancement [25] with the low-precision SRResNet model [8].

Table 3: SoTA image quality (PSNR/SSIM) achieved by the proposed POCA-SWD algorithm with 8-bit and 6-bit model precision, outperforms the recent SR-PTQ [25].

Method	Precision	Sparsity	Set5 ($\times 4$)	Set14 ($\times 4$)	BSD100 ($\times 4$)	Urban100 ($\times 4$)
SRResNet Baseline [25]	FP32 (Baseline)	N/A	32.234 / 0.896	28.656 / 0.784	27.63 / 0.738	26.229 / 0.791
SRResNet [8]			32.05 / 0.901	28.49 / 0.818	27.58 / 0.762	Not Reported
SRResNet (This work)			31.927 / 0.901	28.588 / 0.799	27.628 / 0.756	26.223 / 0.800
AdaRound [18] + LSQ [2]	W8A8	Dense	31.870 / 0.899	28.550 / 0.798	27.533 / 0.754	26.078 / 0.797
SR-PTQ [25]		Dense	32.207 / 0.895	28.619 / 0.783	27.618 / 0.738	26.191 / 0.790
POCA		80%	31.898 / 0.900	28.566 / 0.798	27.620 / 0.756	26.195 / 0.799
Quality Drop [2, 18]	W8A8		0.057 / 0.002	0.038 / 0.001	0.095 / 0.002	0.145 / 0.003
Quality Drop [25]			0.027 / 0.001	0.037 / 0.001	0.012 / 0.000	0.038 / 0.001
Quality Drop (POCA)			0.029 / 0.001	0.022 / 0.001	0.008 / 0.000	0.028 / 0.001
AdaRound [18] + LSQ [2]	W6A6	Dense	31.765 / 0.897	28.449 / 0.795	27.470 / 0.752	25.988 / 0.794
SR-PTQ [25]		Dense	32.089 / 0.892	28.504 / 0.779	27.561 / 0.733	26.011 / 0.783
POCA		80%	31.789 / 0.899	28.499 / 0.796	27.565 / 0.753	26.110 / 0.795
Quality Drop [2, 18]	W6A6		0.162 / 0.004	0.139 / 0.003	0.158 / 0.004	0.235 / 0.006
Quality Drop [25]			0.145 / 0.004	0.152 / 0.005	0.069 / 0.005	0.218 / 0.008
Quality Drop (POCA)			0.138 / 0.002	0.089 / 0.004	0.061 / 0.003	0.113 / 0.005

With 8-bit and 6-bit precision, the proposed method outperforms [25] in BSD100 and Urban100 datasets, as shown in Table 3. For a fair comparison, we listed all the baseline full-precision models. Our method outperforms [25] on all datasets with the new SoTA image quality. Compared to [18] and [2], the improved image quality implies the effectiveness of the proposed SWD. **More importantly**, our method enables the sparse PTQ by using only **20%** pixels of the selected samples, whereas all prior works [2, 18, 25] require dense PTQ.

6 Conclusion and Discussion

In this paper, we propose a novel post-training quantization algorithm designed for high-quality Codec Avatar rendering. As the first work in this area, we discover the challenge of temporal noise with low-precision avatar decoding. Through thorough investigation, we reveal the root cause of the temporal noise that originated from the high dynamics and long-tailedness of the texture distribution. To address that, we propose POCA, which consists of view-aware PTQ and shape-wise filtering (SWD). Compared to the previous SoTA methods, POCA eliminates the temporal noises even with 6-bit precision. Furthermore, the proposed SWD algorithm is also applicable to super-resolution tasks, achieving the new SoTA image quality with highly efficient sparse calibration.

Limitations. The proposed POCA algorithm mainly focuses on the VAE-based Codec Avatar generation, which has been proven as a successful strategy with the actual commercial-level VR product deployment. Recently, 3D Gaussian-based avatar rendering also emerged as a feasible method for generative Avatars, while low-precision quantization of such models will remain as future work.

References

1. Choi, J., Venkataramani, S., Srinivasan, V.V., Gopalakrishnan, K., Wang, Z., Chuang, P.: Accurate and efficient 2-bit quantized neural networks. *Proceedings of Machine Learning and Systems* **1**, 348–359 (2019) [5](#)
2. Esser, S.K., McKinstry, J.L., Bablani, D., Appuswamy, R., Modha, D.S.: Learned step size quantization. In: *International Conference on Learning Representations (ICLR)* (2020) [2](#), [3](#), [5](#), [6](#), [8](#), [11](#), [12](#), [14](#)
3. Fu, Y., Li, Y., Li, C., Saragih, J., Zhang, P., Dai, X., Lin, Y.C.: Auto-card: Efficient and robust codec avatar driving for real-time mobile telepresence. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023) [4](#)
4. Gorbachev, Y., Fedorov, M., Slavutin, I., Tugarev, A., Fatekhov, M., Tarkan, Y.: Openvino deep learning workbench: Comprehensive analysis and tuning of neural networks inference. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (CVPR-W)* (2019) [4](#)
5. Han, Y., Yang, S., Wang, W., Liu, J.: From design draft to real attire: Unaligned fashion image translation. In: *Proceedings of the 28th ACM International Conference on Multimedia*. p. 1533–1541 (2020) [4](#)
6. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013) [4](#)
7. Le, H., Zhang, L., Said, A., Sautiere, G., Yang, Y., Shrestha, P., Yin, F., Pourreza, R., Wiggers, A.: Mobilecodec: neural inter-frame video compression on mobile devices. In: *Proceedings of the 13th ACM Multimedia Systems Conference*. pp. 324–330 (2022) [5](#)
8. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4681–4690 (2017) [3](#), [14](#)
9. Lewis, J.P., Anjyo, K., Rhee, T., Zhang, M., Pighin, F.H., Deng, Z.: Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)* **1**(8), 2 (2014) [4](#)
10. Li, Y., Dong, X., Wang, W.: Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks. In: *International Conference on Learning Representations (ICLR)* (2020) [5](#)
11. Li, Y., Gong, R., Tan, X., Yang, Y., Hu, P., Zhang, Q., Yu, F., Wang, W., Gu, S.: {BRECQ}: Pushing the limit of post-training quantization by block reconstruction. In: *International Conference on Learning Representations (ICLR)* (2021) [2](#), [5](#)
12. 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 (CVPR)* (2023) [3](#), [5](#), [6](#), [13](#)
13. Liu, Z., Cheng, K.T., Huang, D., Xing, E.P., Shen, Z.: Nonuniform-to-uniform quantization: Towards accurate quantization via generalized straight-through estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022) [5](#)
14. Lombardi, S., Saragih, J., Simon, T., Sheikh, Y.: Deep appearance models for face rendering. *ACM Transactions on Graphics (ToG)* (2018) [1](#), [2](#), [4](#), [7](#), [13](#)

15. Ma, S., Simon, T., Saragih, J., Wang, D., Li, Y., De La Torre, F., Sheikh, Y.: Pixel codec avatars. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) [1](#), [2](#), [4](#), [13](#)
16. Mantiuk, R.K., Denes, G., Chapiro, A., Kaplanyan, A., Rufo, G., Bachy, R., Lian, T., Patney, A.: FovVideoVDP: A visible difference predictor for wide field-of-view video. In: ACM Transactions on Graphics (TOG). ACM New York, NY, USA (2021) [3](#), [6](#), [11](#), [12](#)
17. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) [5](#)
18. Nagel, M., Amjad, R.A., Van Baalen, M., Louizos, C., Blankevoort, T.: Up or down? adaptive rounding for post-training quantization. In: International Conference on Machine Learning (ICML) (2020) [2](#), [3](#), [5](#), [6](#), [8](#), [11](#), [12](#), [14](#)
19. Park, E., Yoo, S.: Profit: A novel training method for sub-4-bit mobilenet models. In: European Conference on Computer Vision (ECCV) (2020) [6](#)
20. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems (NeurIPS) (2019) [3](#), [11](#), [12](#), [13](#)
21. van Rozendaal, T., Singhal, T., Le, H., Sautiere, G., Said, A., Buska, K., Raha, A., Kalatzis, D., Mehta, H., Mayer, F., et al.: Mobilenvc: Real-time 1080p neural video compression on a mobile device. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 4323–4333 (2024) [5](#)
22. Saito, S., Schwartz, G., Simon, T., Li, J., Nam, G.: Relightable gaussian codec avatars. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 130–141 (2024) [2](#)
23. Schwartz, G., Wei, S.E., Wang, T.L., Lombardi, S., Simon, T., Saragih, J., Sheikh, Y.: The eyes have it: An integrated eye and face model for photorealistic facial animation. ACM Transactions on Graphics (TOG) (2020) [1](#)
24. Siddegowda, S., Fournarakis, M., Nagel, M., Blankevoort, T., Patel, C., Khobare, A.: Neural network quantization with AI model efficiency toolkit (AIMET). arXiv preprint arXiv:2201.08442 (2022) [2](#), [3](#), [4](#), [11](#)
25. Tu, Z., Hu, J., Chen, H., Wang, Y.: Toward accurate post-training quantization for image super resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023) [2](#), [3](#), [6](#), [14](#)
26. Wang, Z., Wu, Z., Lu, J., Zhou, J.: Bidet: An efficient binarized object detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) (2020) [2](#)
27. Wei, S.E., Saragih, J., Simon, T., Harley, A.W., Lombardi, S., Perdoch, M., Hypes, A., Wang, D., Badino, H., Sheikh, Y.: VR facial animation via multiview image translation. ACM Transactions on Graphics (TOG) (2019) [1](#)
28. Wei, X., Gong, R., Li, Y., Liu, X., Yu, F.: QDrop: Randomly dropping quantization for extremely low-bit post-training quantization. In: International Conference on Learning Representations (ICLR) (2022) [2](#), [5](#), [13](#)
29. Wu, C.h., Zheng, N., Ardisson, S., Bali, R., Belko, D., Brockmeyer, E., Evans, L., Godisart, T., Ha, H., Huang, X., Hypes, A., Koska, T., Krenn, S., Lombardi, S., Luo, X., McPhail, K., Millerschoen, L., Perdoch, M., Pitts, M., Richard, A., Saragih, J., Saragih, J., Shiratori, T., Simon, T., Stewart, M., Trimble, A., Weng, X., Whitewolf, D., Wu, C., Yu, S.I., Sheikh, Y.: Multiface: A dataset for neural

- face rendering. In: arXiv (2022). <https://doi.org/10.48550/ARXIV.2207.11243>, <https://arxiv.org/abs/2207.11243> 1, 11, 12
30. 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 (ICML) (2023) 2, 5, 13
 31. Yin, H., Gong, Y., Qiu, G.: Side window filtering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 13
 32. Zhang, Y., Zhang, Z., Lew, L.: Pokebnn: A binary pursuit of lightweight accuracy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 4