Gradient Knows Best: Mixed-Precision Quantization via Gradient-Guided Bit Allocation for Super-Resolution

Supplementary Material

We provided additional information and analyses, which are listed below:

- **Appendix A**: We provided additional implementation details.
- **Appendix B**: We included additional analysis on activation variance and quantization sensitivity.
- **Appendix C**: We presented additional experimental results, both quantitative and qualitative.
- **Appendix D**: We provided additional ablation studies.

A. Additional implementation details

Implementation details This study implemented the super-resolution (SR) models [2, 13, 16, 17, 25] by quantizing both the weights and the activations. The entire training procedure consisted of three stages: (1) bound initialization, (2) gradient-guided bit allocation (GBA), and (3) bit-aware fine-tuning. In the first stage, bound initialization was performed for one epoch with a batch size of 16. The input image patch size was fixed at 384×384. In this stage, the quantization bounds for activations, l_k^a and u_k^a, were initialized using the MinMax method [11] and then continuously updated via an exponential moving average (EMA) with decay rate $\beta = 0.9$. Similarly, the quantization bounds of weights, l_k^w and u_k^w, were initialized using the optimal mean squared error (OMSE) method [6]. All continuous bit offset parameters s_k were initialized to zero. In the second stage, GBA was conducted with fixed weights. GBA consisted of two sub-stages, each dedicated to computing the gradients of bit offsets g_k^a and g_k^w for activations and weights, respectively. Each sub-stage ran for two epochs with a batch size of 2, and the initial learning rate was set to 0.1. The third stage, bit-aware fine-tuning, was also conducted for two epochs with a batch size of 2. In this phase, the quantization ranges for both activations and weights (l_k^a, u_k^a and l_k^w, u_k^w) were treated as learnable parameters and updated independently during training with a learning rate of 0.01. Additionally, we compared two settings: one applying mixed-precision quantization (MPQ) only to activations, and another applying MPQ to both activations and weights. All experiments were implemented using the PyTorch framework [21] in Python and executed on an NVIDIA RTX 3090 GPU.

B. Additional analyses

Sample-wise and channel-wise variance Figure 1 presents a quantitative visualization of channel-wise activation variance per layer and channel-wise activation variance across input samples. The left part of Fig. 1 shows the per-channel variance of activations in each layer of the SR model, computed from a single input sample, supporting the need for dynamic activation range normalization (DAN). For instance, in the sixth layer, channels 6 and 7 exhibit activation variances of 1656 and 55, respectively, indicating a variance disparity exceeding 30 times. This imbalance in channel-wise activation variance can result in suboptimal quantization. When a fixed quantization range is uniformly applied across channels, channels with high variance may suffer from excessive clipping, whereas channels with low variance may not fully utilize the available precision. The right part of Fig. 1 illustrates how the activation variance of each channel within a single layer varies across different input image samples. Even within the same channel, the variance can range from several tens to several hundreds depending on the input. For example, some channels exhibit activation variances as high as 786 for certain samples, while for other samples, the activation variances drop below 50, indicating substantial sample-dependent fluctuation. In other words, even within the same channel, the activation distribution can vary significantly depending on the input, making it difficult to effectively handle all samples using a fixed normalization range. The channel-wise imbalance and sample-dependent variability in activation distributions are key factors that degrade quantization performance and training stability in SR models where batch normalization (BN) [10] has been removed [17]. DAN addresses these issues by dynamically normalizing the activation distribu-

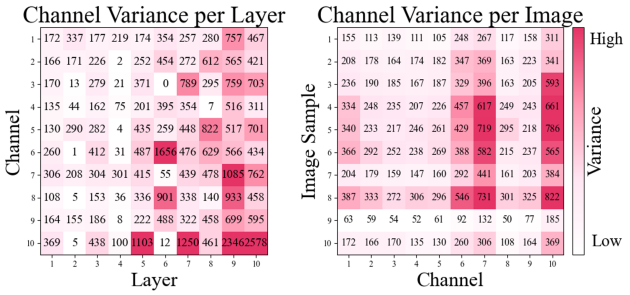

Figure 1. The heatmaps visualize activation variance in randomly selected layers and channels of the EDSR ×4 model after passing 10 randomly selected images from Urban100. Darker colors indicate higher variance. The left map shows large variance differences across channels, while the right map reveals significant variance fluctuations within channels depending on the input image samples.

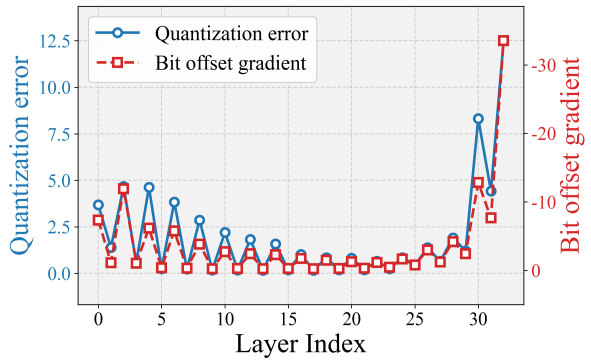

Figure 2. Layer-wise activation quantization is analyzed using two metrics in the EDSR model. The blue line shows the quantization error, i.e., the MSE between the original output and the output after quantizing only one layer's activation. The red line represents the sensitivity estimated by the gradient of the activation bit offset parameter for each layer.

tion of each channel by input sample. By reducing both inter-channel and inter-sample variance disparities, DAN effectively minimizes quantization error and significantly improves overall training stability.

Quantization sensitivity estimation Figure 2 compares the effects of activation quantization across different layers of the EDSR [17] model using two distinct evaluation metrics. The blue curve shows the mean squared error (MSE) between the model's original output and the output after quantizing the activations of a single layer to 4-bit precision, with all other layers kept at 32-bit. The red curve represents the quantization sensitivity estimated using GBA, which leverages the magnitude of the gradient g_k with respect to the bit offset parameters s_k of each layer k. In GBA, smaller gradient magnitudes are interpreted as indicating greater quantization sensitivity for the corresponding layer. Accordingly, as derived from Eq. (6) in the main text, the sign of the gradient values is inverted to enable a direct comparison with the quantization error. GBA quantitatively estimates the quantization sensitivity of each layer by measuring the magnitude and direction of the gradient g_k propagated to the continuous bit offset parameter s_k of each layer with respect to the loss function L_{grad} during backpropagation. A small value of g_k indicates that the currently assigned bit-width is insufficient to reduce the loss, suggesting that the corresponding layer requires higher precision. Based on this gradient-driven sensitivity information, GBA adaptively increases or decreases the bit-width of each layer to perform efficient MPQ. The results in Fig. 2 show that the sensitivity estimated by GBA closely aligns with the actual quantization error, clearly demonstrating that GBA reliably captures the layer-wise impact of quantization.

Theoretical basis of GBA This section provides a mathematical proof that a small gradient in the negative direction during backpropagation is not merely a heuristic indication of insufficient bit-width, but a direct outcome derived from the chain-rule structure of fake quantization [11]. In this context, a small gradient in the negative direction with respect to the bit-width indicates that increasing the bit-width would considerably reduce the loss, suggesting that the current precision is insufficient and results in large quantization errors. Therefore, such a gradient serves as a direct signal that more bit-width should be allocated to that layer.

As defined in Eq. (2) of the main text, the quantization step Δ_k depends on the bit-width b_k, where k denotes the layer index. To be consistent with the definition of the layer-wise gradient in Eq. (6) of the main text, g_k reflects the averaged bit-width sensitivity accumulated across the calibration samples. For the loss L_{grad} in Eq. (3), the gradient with respect to the bit parameter s_k is computed using the chain-rule as follows:

$$\frac{\partial L_{grad}}{\partial s_k} = \frac{\partial L_{grad}}{\partial b_k} \cdot \frac{\partial b_k}{\partial s_k}. \tag{S1}$$

Since $\frac{\partial b_k}{\partial s_k} > 0$, the sign of the gradient $g_k = \frac{\partial L_{grad}}{\partial s_k}$ is identical to the sign that determines the direction of $\frac{\partial L_{grad}}{\partial b_k}$.

Considering the influence of b_k on the quantization step Δ_k, the derivative can be written as follows:

$$\frac{\partial L_{grad}}{\partial b_k} = \frac{\partial L_{grad}}{\partial \Delta_k} \cdot \frac{\partial \Delta_k}{\partial b_k}. \tag{S2}$$

As the quantization step increases, the quantization error increases ($\frac{\partial L_{grad}}{\partial \Delta_k} > 0$), while increasing the bit-width decreases the step size ($\frac{\partial \Delta_k}{\partial b_k} < 0$). Therefore, the resulting expression is derived as follows:

$$\frac{\partial L_{grad}}{\partial b_k} < 0 \quad \Rightarrow \quad g_k = \frac{\partial L_{grad}}{\partial s_k} < 0. \tag{S3}$$

Thus, $g_k < 0$ mathematically indicates that increasing the bit-width reduces the loss. Therefore, GBA effectively estimates the quantization sensitivity of each layer based on the direct derivative of the loss with respect to the bit-width and incorporates bit precision as a learnable differentiable parameter within the optimization procedure. These derivations provide a theoretical foundation for gradient-driven bit adjustment.

Methodological novelty and effectiveness of GBA Although gradient information has been utilized in various quantization frameworks, the interpretation and integration of gradient information into precision optimization differ substantially across existing methods. From the perspective of MPQ, this section highlights the methodological novelty and practical effectiveness of the proposed GBA framework compared to representative gradient-based approaches.

Table 1. Performance comparison with PTQ-based static quantization methods, all applied with fine-tuning, on the EDSR and RDN models for $\times 2$ **scaling** (which is different from the $\times 4$ scaling used in the main text). W/A denotes the bit precision for weights and activations, respectively, and MP indicates whether mixed precision was applied.

Method	Venue	W/A	Processing Time	Set5		Set14		BSD100		Urban100		Manga109	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EDSR	CVPRW'17	32 / 32	–	37.99	0.961	33.57	0.917	32.16	0.900	31.98	0.987	38.55	0.977
EDSR-MinMax+FT	–	4 / 4	49 sec	35.37	0.918	32.07	0.878	30.95	0.859	30.14	0.881	35.42	0.934
EDSR-Percentile+FT	–	4 / 4	67 sec	27.93	0.893	27.97	0.876	28.22	0.869	25.47	0.857	22.75	0.918
EDSR-PTQ4SR	CVPR'23	4 / 4	127 sec	36.73	0.945	32.76	0.901	31.55	0.883	30.82	0.907	36.46	0.959
EDSR-AdaBM	CVPR'24	4 / 4MP	50 sec	37.18	0.952	33.01	0.908	31.65	0.888	31.08	0.913	37.34	**0.978**
EDSR-AdaBM*	CVPR'24	4MP / 4MP	50 sec	37.16	0.952	33.01	0.908	31.65	0.888	31.09	**0.918**	37.37	0.969
EDSR-Ours	–	4 / 4MP	26 sec	37.59	0.957	33.22	0.912	31.92	**0.895**	31.32	0.916	37.56	0.969
EDSR-Ours*	–	4MP / 4MP	26 sec	**37.65**	**0.958**	**33.25**	**0.913**	**31.97**	**0.895**	**31.43**	**0.918**	**37.73**	0.971
EDSR-MinMax+FT	–	3 / 3	49 sec	34.20	0.926	31.00	0.880	30.14	0.861	27.64	0.852	32.13	0.935
EDSR-Percentile+FT	–	3 / 3	67 sec	28.23	0.892	27.98	0.871	28.17	0.863	25.41	0.847	22.72	0.913
EDSR-PTQ4SR	CVPR'23	3 / 3	127 sec	34.72	0.941	31.19	0.891	30.32	0.871	27.69	0.866	32.20	0.948
EDSR-AdaBM	CVPR'24	3 / 3MP	50 sec	35.51	0.932	31.92	0.886	30.75	0.863	29.24	0.879	34.71	0.944
EDSR-AdaBM*	CVPR'24	3MP / 3MP	50 sec	34.84	0.941	31.30	0.892	30.38	0.871	27.85	0.868	32.58	0.950
EDSR-Ours	–	3 / 3MP	26 sec	36.68	0.953	32.57	0.905	31.41	0.887	30.02	0.901	36.18	0.964
EDSR-Ours*	–	3MP / 3MP	26 sec	**37.10**	**0.955**	**32.86**	**0.908**	**31.64**	**0.891**	**30.60**	**0.908**	**36.88**	**0.966**
RDN	CVPR'18	32 / 32	–	38.05	0.961	33.64	0.918	32.19	0.900	32.22	0.930	38.59	0.977
RDN-MinMax+FT	–	4 / 4	162 sec	32.40	0.882	29.89	0.836	29.28	0.815	27.05	0.810	30.51	0.892
RDN-Percentile+FT	–	4 / 4	462 sec	23.84	0.616	23.74	0.660	25.07	0.649	21.56	0.601	17.46	0.583
RDN-PTQ4SR	CVPR'23	4 / 4	757 sec	33.43	0.927	30.40	0.875	29.65	0.854	27.07	0.846	30.32	0.920
RDN-AdaBM	CVPR'24	4 / 4 MP	167 sec	33.86	0.931	30.75	0.881	29.89	0.858	27.40	0.856	31.81	0.939
RDN-AdaBM*	CVPR'24	4MP / 4MP	167 sec	33.86	0.930	30.69	0.878	29.84	0.855	27.32	0.852	31.81	0.939
RDN-Ours	–	4 / 4MP	87 sec	**37.65**	**0.958**	**33.35**	**0.912**	**32.00**	**0.895**	**31.65**	**0.919**	**37.61**	**0.970**
RDN-Ours*	–	4MP / 4MP	87 sec	37.16	0.954	32.87	0.908	31.69	0.891	30.56	0.907	36.59	0.965
RDN-MinMax+FT	–	3 / 3	162 sec	31.26	0.885	28.66	0.817	28.30	0.791	25.62	0.786	28.26	0.884
RDN-Percentile+FT	–	3 / 3	462 sec	24.21	0.909	24.16	0.853	25.25	0.841	21.79	0.820	17.78	0.878
RDN-PTQ4SR	CVPR'23	3 / 3	757 sec	33.32	0.931	30.26	0.876	29.57	0.857	26.83	0.845	29.78	0.920
RDN-AdaBM	CVPR'24	3 / 3MP	167 sec	33.68	0.928	30.55	0.880	29.82	0.859	27.16	0.854	31.40	0.938
RDN-AdaBM*	CVPR'24	3MP / 3MP	167 sec	33.71	0.932	30.60	0.881	29.80	0.859	27.18	0.853	31.41	0.938
RDN-Ours	–	3 / 3MP	87 sec	**36.76**	**0.951**	**32.59**	0.900	**31.52**	0.885	**30.22**	**0.896**	35.73	0.953
RDN-Ours*	–	3MP / 3MP	87 sec	36.64	0.949	32.58	**0.902**	31.47	**0.886**	30.10	0.895	**35.83**	**0.954**

- **GradQ-ViT** [5]: This method is designed for image classification and focuses on improving training stability by quantizing gradients during backpropagation. It does not consider quantization sensitivity or mixed-precision bit allocation, and its primary objective is to enhance training efficiency rather than to optimize precision assignment.
- **Chauhan et al.** [4]: This method perturbs pretrained weights within an ϵ-neighborhood and assigns bit-widths through complex integer linear programming (ILP) based on the expected gradient norm at the perturbed points. However, such perturbation-based estimation cannot accurately reflect the discrete distortion characteristics induced by real quantization, making it difficult to precisely approximate actual quantization error. In addition, although gradient information is utilized, its interpretation and intended role are fundamentally different from those in GBA.
- **BMPQ** [12]: This method is based on QAT and incurs significantly high training overhead. It indirectly estimates layer sensitivity by statistically approximating the magnitude of weight gradients ($\partial L/\partial W$) and assigns bit-widths through ILP-based global optimization. However, it uses the magnitude of weight gradients merely as a surrogate measure for sensitivity, and does not directly incorporate the effect of bit-precision changes on the loss. Furthermore, it fails to capture inter-layer correlations, making it difficult to guarantee globally optimal bit allocation.

Prior mixed-precision approaches performed bit allocation solely in the **discrete integer bit-width space**, where bit-widths are inherently **non-differentiable**, making it im-

Table 2. Performance comparison with PTQ-based **6-bit** quantization methods applied to the EDSR and RDN models with $\times 4$ scaling, which differs from the 4-bit and 3-bit quantization settings used in the main text. W/A denotes the bit precision for weights and activations, respectively, and MP indicates whether mixed precision was applied.

Method	Venue	W/A	Processing Time	Set5		Set14		BSD100		Urban100		Manga109	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EDSR	CVPRW'17	32 / 32	–	32.10	0.894	28.58	0.781	27.56	0.736	26.04	0.785	30.35	0.907
EDSR-MinMax+FT	–	6 / 6	49 sec	31.89	0.882	28.43	0.773	27.45	0.728	25.89	0.774	30.10	0.895
EDSR-Percentile+FT	–	6 / 6	67 sec	24.77	0.784	24.44	0.712	24.78	0.685	22.08	0.674	19.97	0.781
EDSR-PTQ4SR	CVPR'23	6 / 6	127 sec	31.93	0.887	28.44	0.777	27.48	0.732	25.82	0.777	29.83	0.898
EDSR-AdaBM	CVPR'24	6 / 6 MP	50 sec	31.97	0.888	28.48	0.778	27.48	0.732	25.90	0.778	30.14	0.901
EDSR-AdaBM*	CVPR'24	6MP / 6MP	50 sec	31.92	0.887	28.46	0.777	27.48	0.732	25.88	0.778	30.02	0.899
EDSR-Ours	–	6 / 6 MP	26 sec	32.02	**0.891**	28.53	**0.780**	27.53	**0.735**	25.98	0.782	30.25	0.903
EDSR-Ours*	–	6MP / 6MP	26 sec	**32.06**	**0.891**	**28.54**	**0.780**	**27.54**	**0.735**	**26.00**	**0.783**	**30.27**	**0.904**
RDN	CVPR'18	32 / 32	–	32.24	0.895	28.67	0.784	27.63	0.739	26.29	0.793	30.63	0.911
RDN-MinMax+FT	–	6 / 6	162 sec	30.70	0.853	27.74	0.746	26.90	0.700	25.01	0.731	28.71	0.862
RDN-Percentile+FT	–	6 / 6	462 sec	20.62	0.798	20.09	0.688	21.37	0.669	18.09	0.648	14.58	0.719
RDN-PTQ4SR	CVPR'23	6 / 6	757 sec	31.48	0.882	28.12	0.772	27.23	0.727	25.43	0.766	28.84	0.891
RDN-AdaBM	CVPR'24	6 / 6MP	167 sec	31.65	0.881	28.25	0.770	27.28	0.723	25.52	0.763	29.51	0.891
RDN-AdaBM*	CVPR'24	6MP / 6MP	167 sec	31.63	0.882	28.22	0.770	27.27	0.724	25.48	0.762	29.36	0.889
RDN-Ours	–	6 / 6MP	87 sec	32.18	**0.894**	**28.64**	**0.783**	**27.61**	**0.738**	**26.25**	**0.791**	**30.52**	**0.908**
RDN-Ours*	–	6MP / 6MP	87 sec	**32.24**	**0.894**	**28.64**	**0.783**	**27.61**	**0.738**	26.23	0.790	30.51	0.907

possible to optimize precision directly using gradient information. Consequently, existing methods relied on **indirect sensitivity estimation**, such as approximating the statistical magnitude of weight gradients or evaluating metrics obtained from perturbing pretrained weights within a small neighborhood, followed by **ILP-based discrete search**. However, such heuristic search procedures fail to accurately reflect the actual change in quantization loss caused by bit-width variation and cannot effectively model **global inter-layer dependencies**, since each layer is optimized independently.

To address these limitations, GBA redefines bit-width as a **learnable continuous parameter** and directly optimizes it using gradients $(\partial L/\partial s)$ computed from the loss function, explicitly modeling the effect of precision adjustments on quantization loss. GBA captures the actual loss change induced by bit-width variation without updating weights and with only a small calibration set. In addition, the bit-widths of different layers are inherently connected through gradient flows, such that a bit change in one layer affects the loss gradients of others, enabling the consideration of **inter-layer dependencies**. As a result, GBA performs joint optimization across the entire network rather than independent layer-wise assignment, allowing global bit allocation to emerge intrinsically through **gradient-based learning** without complex ILP procedures and minimizing performance degradation. Thus, GBA fundamentally overcomes the limitations of prior discrete-search-based approaches by enabling **differentiable precision optimization** in a **continuous space**.

C. Additional experimental results

Comparison on scaling factor of 2 In addition to the $\times 4$ scale SR network evaluation in the main text, we further evaluated our method on the $\times 2$ scale setting. Table 1 compares the proposed method with existing post-training quantization (PTQ)-based approaches [8, 11, 14, 22] applied to the EDSR [17] and RDN [25] models under the $\times 2$ scale configuration. For a fair comparison, we followed prior studies [8, 22] by keeping the first and last layers fixed to 8-bit precision. In the table, the $\star$ symbol indicates a modified version of AdaBM [8], where mixed precision (MP) is applied to both weights and activations, as opposed to its default configuration, which applies MP only to activations. To ensure a fair comparison, we also included the results of AdaBM* with MP applied to both weights and activations. The proposed method consistently outperformed existing PTQ-based techniques in terms of peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). Notably, under the 3-bit quantization setting, our method (RDN-Ours*) achieved a PSNR of 30.10dB on Urban100 [9], which was 2.92dB higher than AdaBM* (27.18dB) under the same conditions.

Comparison on 6-bit quantization The proposed method consistently achieved high performance not only in low-bit settings but also under higher-bit configurations.

Table 3. Performance comparison of PTQ-based quantization methods applied to the **CARN model** with $\times 4$ scaling, which was not included in the main text where only the EDSR and RDN models were evaluated. W/A denotes the bit precision for weights and activations, respectively, and MP indicates whether mixed precision was applied.

Method	Venue	W/A	Processing Time	Set5		Set14		BSD100		Urban100		Manga109	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CARN	ECCV'18	32 / 32	–	31.95	0.891	28.51	0.780	27.51	0.734	25.82	0.777	30.08	0.904
CARN-MinMax+FT	–	4 / 4	36 sec	24.12	0.726	23.53	0.661	23.12	0.604	21.94	0.646	23.52	0.767
CARN-Percentile+FT	–	4 / 4	69 sec	18.41	0.683	17.37	0.581	19.34	0.491	16.33	0.548	12.49	0.601
CARN-PTQ4SR	CVPR'23	4 / 4	96 sec	28.45	0.782	26.39	0.701	25.71	0.658	23.91	0.679	26.62	0.795
CARN-AdaBM	CVPR'24	4 / 4MP	38 sec	29.47	0.817	26.89	0.715	25.94	0.661	24.21	0.695	27.16	0.822
CARN-AdaBM*	CVPR'24	4MP / 4MP	38 sec	29.43	0.815	26.87	0.715	25.92	0.662	24.29	0.698	27.42	0.825
CARN-Ours	–	4 / 4MP	19 sec	**31.41**	**0.880**	**28.16**	0.766	**27.24**	0.723	**25.32**	**0.755**	**28.94**	**0.878**
CARN-Ours*	–	4MP / 4MP	19 sec	31.38	**0.880**	28.07	**0.767**	**27.24**	**0.724**	**25.32**	0.754	28.85	0.877
CARN-MinMax+FT	–	3 / 3	36 sec	18.59	0.570	18.69	0.531	18.86	0.534	17.61	0.326	18.65	0.409
CARN-Percentile+FT	–	3 / 3	69 sec	17.58	0.540	16.76	0.542	18.78	0.543	15.89	0.514	12.05	0.551
CARN-PTQ4SR	CVPR'23	3 / 3	96 sec	23.29	0.625	22.89	0.596	21.78	0.554	20.81	0.559	22.95	0.661
CARN-AdaBM	CVPR'24	3 / 3MP	38 sec	25.52	0.682	24.15	0.607	23.77	0.564	22.04	0.575	23.47	0.672
CARN-AdaBM*	CVPR'24	3MP / 3MP	38 sec	25.31	0.672	23.98	0.592	23.37	0.552	21.83	0.567	23.22	0.651
CARN-Ours	–	3 / 3MP	19 sec	29.51	0.848	26.77	0.736	26.63	0.700	24.21	0.707	26.23	0.820
CARN-Ours*	–	3MP / 3MP	19 sec	**30.42**	**0.854**	**27.48**	**0.744**	**26.85**	**0.706**	**24.50**	**0.715**	**27.26**	**0.835**

Table 4. Performance comparison of PTQ-based quantization methods applied to the **SRResNet model** with $\times 4$ scaling, which was not included in the main text where only the EDSR and RDN models were evaluated. W/A denotes the bit precision for weights and activations, respectively, and MP indicates whether mixed precision was applied.

Method	Venue	W/A	Processing Time	Set5		Set14		BSD100		Urban100		Manga109	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SRResNet	CVPR'17	32 / 32	–	31.86	0.889	28.42	0.778	27.46	0.733	25.67	0.772	29.84	0.900
SRResNet-MinMax+FT	–	4 / 4	48 sec	30.54	0.861	27.20	0.751	26.74	0.707	24.73	0.730	27.07	0.860
SRResNet-Percentile+FT	–	4 / 4	90 sec	21.87	0.820	21.17	0.708	21.98	0.684	19.06	0.692	15.28	0.758
SRResNet-PTQ4SR	CVPR'23	4 / 4	110 sec	30.96	0.868	27.85	0.757	27.01	0.712	24.90	0.736	28.23	0.868
SRResNet-AdaBM	CVPR'24	4 / 4MP	50 sec	31.37	0.875	28.03	0.762	27.07	0.714	25.17	0.746	28.96	0.879
SRResNet-AdaBM*	CVPR'24	4MP / 4MP	50 sec	31.36	0.875	28.08	0.763	27.16	0.718	25.22	0.748	29.07	0.881
SRResNet-Ours	–	4 / 4MP	24 sec	31.51	0.882	28.21	**0.771**	**27.33**	**0.728**	25.44	0.761	29.26	0.889
SRResNet-Ours*	–	4MP / 4MP	24 sec	**31.65**	**0.884**	**28.27**	**0.771**	**27.33**	0.727	**25.50**	**0.763**	**29.50**	**0.892**
SRResNet-MinMax+FT	–	3 / 3	48 sec	29.22	0.823	26.66	0.719	26.21	0.680	23.78	0.681	25.30	0.800
SRResNet-Percentile+FT	–	3 / 3	90 sec	21.64	0.799	21.04	0.693	21.84	0.669	18.85	0.661	15.16	0.729
SRResNet-PTQ4SR	CVPR'23	3 / 3	110 sec	29.69	0.830	26.99	0.726	26.45	0.684	24.04	0.689	26.54	0.816
SRResNet-AdaBM	CVPR'24	3 / 3MP	50 sec	30.11	0.847	27.28	0.739	26.66	0.676	24.25	0.703	27.01	0.834
SRResNet-AdaBM*	CVPR'24	3MP / 3MP	50 sec	30.06	0.846	27.26	0.739	26.63	0.695	24.22	0.701	26.96	0.834
SRResNet-Ours	–	3 / 3MP	24 sec	30.89	0.870	27.80	**0.760**	27.06	0.718	24.89	0.740	28.28	0.870
SRResNet-Ours*	–	3MP / 3MP	24 sec	**31.06**	**0.874**	**27.88**	**0.760**	**27.10**	**0.719**	**25.01**	**0.744**	**28.38**	**0.872**

In particular, since the previous study AdaBM [8] reported results for both 4-bit and 6-bit settings, we additionally conducted 6-bit quantization experiments under the same conditions for a fair comparison. Table 2 summarizes the PSNR and SSIM results when applying 6-bit quantization to the EDSR [17] and RDN [25] models. The results show that the proposed method maintained performance comparable to full-precision even at 6 bits. For example, our method achieved 27.54dB on BSD100 [18], with only a 0.02dB drop compared to the 32-bit baseline. Similarly, our method with the RDN model also demonstrated stable performance across most benchmark datasets, achieving 26.23dB on Urban100 [9] and 30.51dB on Manga109 [19], while maintaining high reconstruction quality even after quantization. These results indicate that the proposed method effectively minimized information loss under

Table 5. Performance comparison of PTQ-based quantization methods applied to the ResNet-18 model for **image classification** under various bit-widths. MPQ indicates whether mixed-precision quantization was applied. Classification accuracy (%) is reported for 4-, 6-, and 8-bit settings, along with the full-precision (FP32) baseline.

Model	Method	Venue	MPQ	Bit-width			FP32
				4	6	8	
ResNet-18	AdaRound	ICML'20	✗	67.26%	70.17%	70.78%	71.00%
	BRECQ	ICLR'21	✗	68.21%	70.44%	70.80%	71.00%
	Qdrop	ICLR'22	✗	**69.54%**	**70.52%**	**70.83%**	71.00%
	PTMQ	AAAI'24	✓	67.57%	70.23%	70.79%	71.00%
	Ours	–	✓	68.33%	70.30%	**70.83%**	71.00%

Table 6. Quantitative results of 4-bit post-training quantization for EDSR ($\times 4$) on Set5 and Urban100 datasets with **varying numbers of calibration images**. # indicates the number of calibration images used. Each value is reported as mean $\pm$ standard deviation over three trials, where calibration images were randomly sampled in each trial.

#	Set5 ($\times 4$)		Urban100 ($\times 4$)	
	PSNR	SSIM	PSNR	SSIM
20	31.58 $\pm$ 0.038	0.880 $\pm$ 0.000	25.56 $\pm$ 0.029	0.760 $\pm$ 0.001
60	31.58 $\pm$ 0.023	0.880 $\pm$ 0.001	25.58 $\pm$ 0.030	0.760 $\pm$ 0.001
100	31.58 $\pm$ 0.029	0.881 $\pm$ 0.001	25.61 $\pm$ 0.011	0.762 $\pm$ 0.001
150	**31.64** $\pm$ 0.026	**0.882** $\pm$ 0.000	25.62 $\pm$ 0.008	0.763 $\pm$ 0.001
300	31.63 $\pm$ 0.005	0.881 $\pm$ 0.000	25.62 $\pm$ 0.005	0.764 $\pm$ 0.002
500	**31.64** $\pm$ 0.005	**0.882** $\pm$ 0.001	**25.64** $\pm$ 0.017	0.764 $\pm$ 0.002
800	31.63 $\pm$ 0.016	**0.882** $\pm$ 0.000	25.62 $\pm$ 0.019	**0.765** $\pm$ 0.002

Table 7. Quantitative results of 4-bit quantized EDSR ($\times 4$) on Set5 and Urban100 datasets with **varying learning rates**.

Learning rate	Set5 ($\times 4$)		Urban100 ($\times 4$)	
	PSNR	SSIM	PSNR	SSIM
0.001	31.60	0.880	25.58	0.760
0.005	31.58	0.878	25.56	0.760
0.01	**31.61**	0.**881**	**25.61**	**0.762**
0.05	31.60	0.880	25.60	0.760

high-bit settings and provided more consistent performance compared to existing PTQ-based approaches [8, 11, 14, 22].

Comparison on additional SR models Although previous experiments in the main text focused on validating our method on EDSR [17] and RDN [25], this section presents extended evaluations on other SR architectures such as CARN [2] and SRResNet [13] to assess the generalizability of the proposed framework. Table 3 and Table 4 show the results of applying the proposed method to the CARN and SRResNet models under 4-bit and 3-bit quantization settings. For fair comparison, the first and last layers were fixed to 8-bit precision, as in prior experiments, while the remaining layers were quantized using flexible bit-width allocation. In the case of CARN, the proposed method maintained stable performance even in low-bit settings. For example, under the 3-bit configuration, our method achieved a PSNR of 30.42dB on Set5 [3], significantly outperforming AdaBM* [8], which recorded 25.31dB under the same setting, by a margin of 5.1dB. This performance gap remained consistent across more complex datasets with abundant high-frequency details such as Urban100 [9] and Manga109 [19], indicating that our method effectively reduced bit usage without compromising SR quality. Furthermore, the total quantization processing time was only 19 seconds, confirming that the proposed method offers a highly practical trade-off between precision and efficiency. A similar trend was observed with SRResNet. Despite its relatively lightweight architecture, the proposed method achieved strong performance across all bit-width settings. For instance, in the 3-bit setting, SRResNet-Ours* achieved 31.06dB on Set5, outperforming AdaBM* by 1dB. Notably, our method also showed clear advantages on Urban100 and Manga109 while maintaining an entire quantization processing time of just 24 seconds. These results demonstrate that the proposed framework is not limited to specific network architectures but can be effectively applied to a wide range of SR models, consistently delivering high accuracy and computational efficiency compared to existing PTQ-based quantization methods [8, 11, 14, 22].

Comparison on ResNet-18 for image classification Although this study focused on SR models, we extended the proposed quantization framework to the image classification task using ResNet-18 [7] to verify its generalizability. Table 5 compares the results of applying various PTQ methods [15, 20, 23, 24] to ResNet-18. The proposed method achieved a classification accuracy comparable to PTMQ [24], a PTQ method based on MPQ. In particular, under the 8-bit setting, it achieved 70.83% accuracy, which was similar to that of widely used existing methods. Under the 6-bit and 4-bit settings, it recorded 70.30% and 68.33%, respectively—showing slight performance drops but still maintaining competitive accuracy compared to other PTQ-based methods. These results demonstrate that although the framework was originally optimized for SR, it could be successfully applied to general vision tasks such as image classification.

Additional qualitative results In addition to the quantitative evaluation, we conducted a qualitative comparison to visually assess the reconstruction quality of different PTQ methods. Due to the large number of output images, only representative examples were included in the main text, with more examples shown in Fig. 3 and Fig. 4. Both figures present 4-bit quantized results using the EDSR [17] and RDN [25] models, respectively. Figure 3 shows qualitative results from the 4-bit quantized EDSR model. While the 4-bit quantized EDSR model generally maintained overall structural consistency across different methods, noticeable differences emerged in fine details such as text clarity, edges, and high-frequency textures. For instance, in Urban100 img083, the proposed method restored the sharpest character contours and clean textures among all methods. In contrast, MinMax [11] introduced strong color noise and structural distortions, while Percentile [14] suffered from blur and detail loss. MinMax+FT and Percentile+FT yielded slight improvements after fine-tuning but still failed to accurately recover fine structures. PTQ4SR [22] and AdaBM [8] produced more stable and visually coherent results, but the proposed method achieved the most faithful visual restoration with superior edge preservation, texture clarity, and noise suppression. Figure 4 presents the results for the 4-bit quantized RDN model. Due to its deep residual architecture, RDN tended to exhibit higher sensitivity to quantization, especially in scenes with repetitive patterns and high-frequency structures. This was evident in Urban100 img023 and img024, where the MinMax [11] method produced severe color noise that disrupted the regular line patterns, resulting in visually unstable outputs. Although Percentile [14] reduced some of these artifacts, it still suffered from texture blurring and color smearing. Fine-tuned versions, MinMax+FT and Percentile+FT, alleviated structural distortions to some extent, but remained limited in preserving uniformity and restoring sharp details. PTQ4SR [22] and AdaBM [8] provided better edge alignment and noise suppression, yielding more visually coherent results. However, in challenging cases such as img027 and img031, which contained repetitive architectural patterns and complex shading, the proposed method achieved the most stable reconstruction. It successfully preserved structural patterns, edge sharpness, and inter-channel consistency, delivering visual quality most closely aligned with the output of the full-precision (FP32) model.

Table 8. Quantitative comparison of 4-bit quantized EDSR (×4) **using different candidate values of** θ_k for bit offset selection. Each candidate value of θ_k represents a discrete set of bit offset values allowed for layer k, defining the range of bit-width adjustments around the base bit b_{base}.

Candidate values of θ_k	Set5 (×4)		Urban100 (×4)	
	PSNR	SSIM	PSNR	SSIM
{-1, 0, 1}	**31.67**	**0.881**	**25.62**	**0.762**
{-2, –1, 0, 1, 2}	31.56	0.880	25.52	**0.762**
{-3, –2, –1, 0, 1, 2, 3}	31.41	0.880	25.38	0.757

D. Additional ablation studies

Effect of calibration data size For a fair comparison, all experiments presented in the main text used 100 randomly sampled images from the DIV2K training dataset [1] as the calibration data, following the same setting as prior methods. In this supplementary section, we conducted an additional ablation study to analyze more precisely the impact of the number of calibration images on quantization performance. Table 6 shows the PSNR and SSIM results on the Set5 [3] and Urban100 [9] datasets when the number of calibration images varied among 20, 60, 100, 150, 300, 500, and 800. The results revealed that increasing the number of calibration samples gradually improved PSNR up to a certain point; however, in the case of Urban100, the performance gain continued beyond 100 images, with only marginal improvements thereafter. On the other hand, as the number of calibration images increased, the overall quantization time grew linearly, introducing a practical trade-off. Accordingly, we fixed the calibration set size to 100 images in all experiments to balance performance and efficiency while ensuring a fair comparison with existing methods. This setting provided a practical compromise, offering stable quantization quality while minimizing unnecessary computational overhead.

Fine-tuning bounds with learning rate We analyzed the effect of the learning rate on performance during the fine-tuning of quantization bounds (i.e., the lower and upper bounds) for weights and activations. This fine-tuning stage

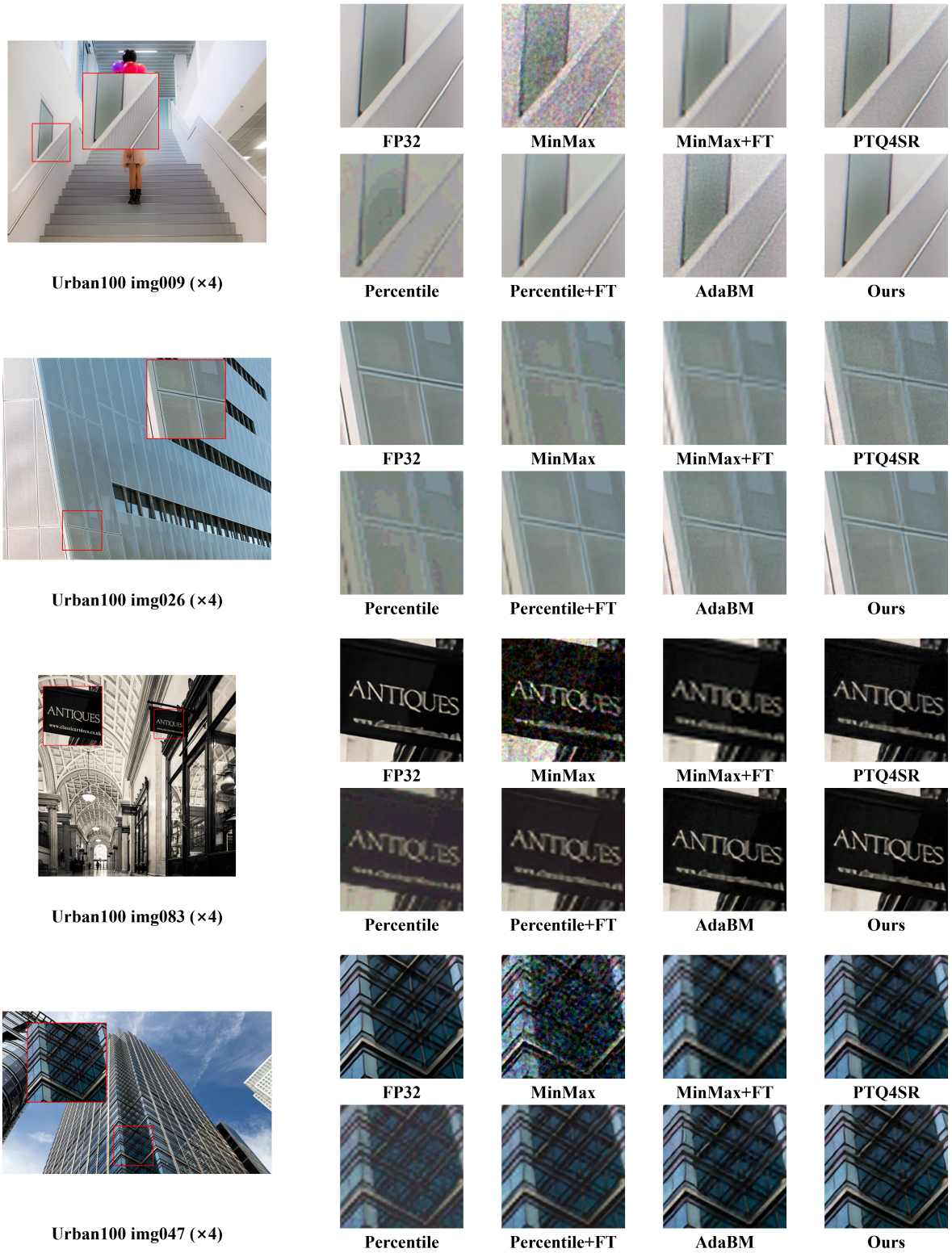

Figure 3. Qualitative results of 4-bit EDSR models quantized by different PTQ methods on Urban100 ($\times 4$ scaling). The left image shows the original, and the right image highlights the region of interest. The cropped images from this region are compared.

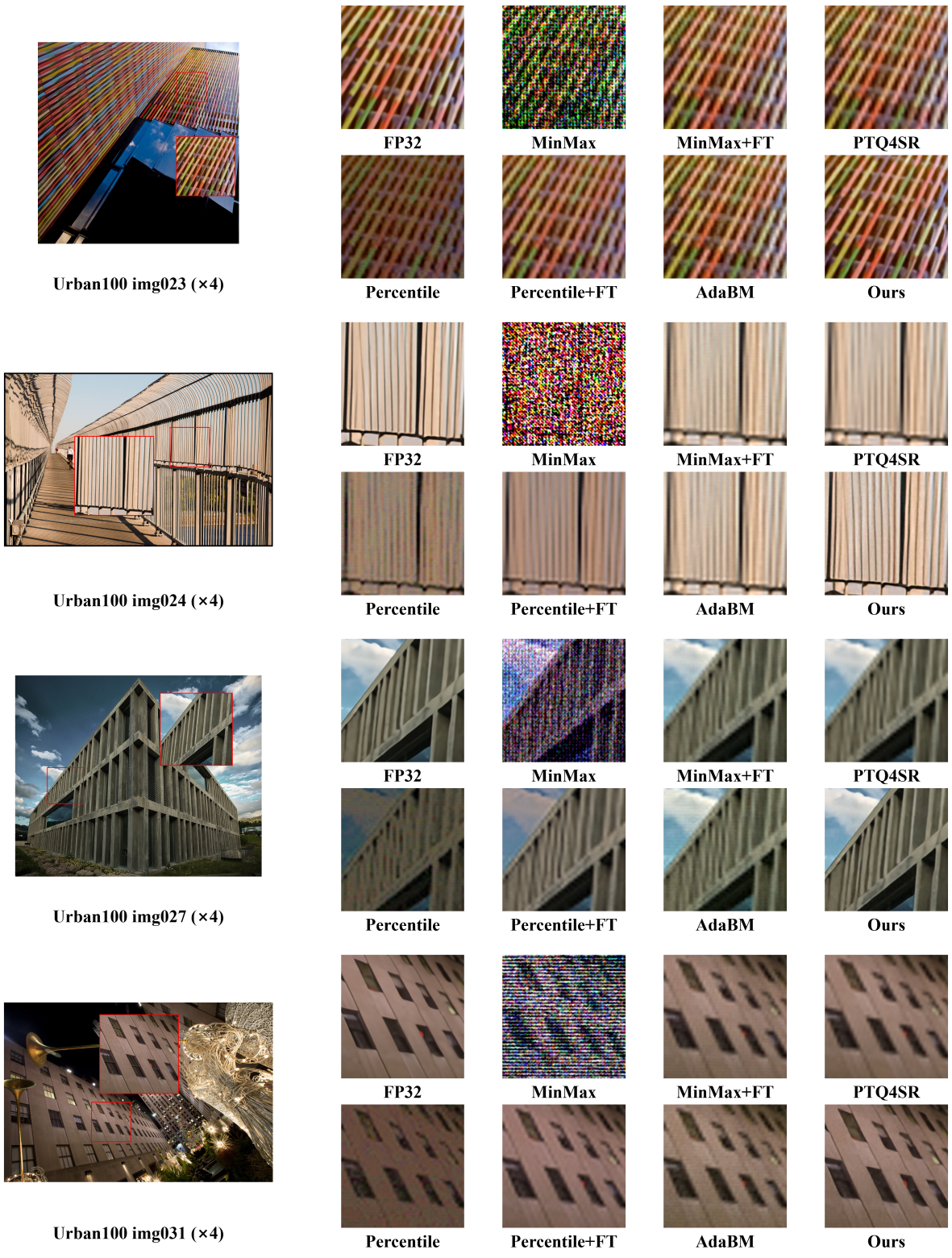

Figure 4. Qualitative results of 4-bit RDN models quantized by different PTQ methods on Urban100 ($\times 4$ scaling). The left image shows the original, and the right image highlights the region of interest. The cropped images from this region are compared.

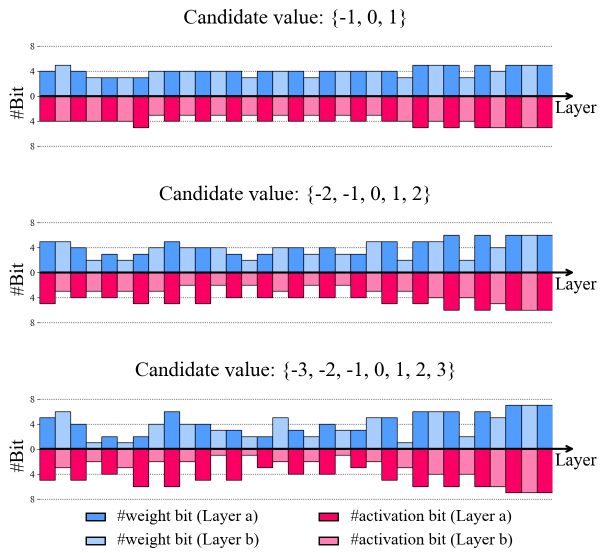

Figure 5. Bit offset values for each layer's weight and activation across three candidate sets. The x-axis represents the layer index, and the y-axis shows the offset from the base bit-width. Each residual block in EDSR has two layers: Layer a is before activation, and Layer b is after (e.g., ReLU). The bar color reflects the data type (weight or activation) and layer stage, visualizing bit-precision variation across the network.

Table 9. Ablation study on the effect of λ_{feat} for 4-bit quantized EDSR ($\times 4$).

λ_{feat}	Set5 ($\times 4$)		Urban100 ($\times 4$)	
	PSNR	SSIM	PSNR	SSIM
0	31.61	0.881	25.61	0.762
5	31.61	0.881	25.61	0.761
10	**31.67**	**0.881**	**25.62**	**0.762**
15	31.66	0.881	25.61	0.761
20	31.66	0.882	25.61	0.761
25	31.63	0.881	25.62	0.760

was conducted over 2 epochs using calibration data, where both rapid convergence and stable improvement in quantization performance were required within a limited number of training iterations. Table 7 presents the PSNR and SSIM results on the Set5 [3] and Urban100 [9] datasets when the initial learning rates were set to 0.001, 0.005, 0.01, and 0.05, respectively, during the bound fine-tuning process. The results showed that all learning rate settings yielded relatively similar performance, but the setting of 0.01 consistently achieved the most stable and superior results across both datasets. Accordingly, all experiments reported in the main text fixed the learning rate of the bound fine-tuning stage to 0.01, allowing effective adjustment of quantization bounds within 2 epochs.

Table 10. Ablation study on the effect of **calibration patch size** for 4-bit quantized EDSR ($\times 4$).

Patch size	Set5 ($\times 4$)		Urban100 ($\times 4$)	
	PSNR	SSIM	PSNR	SSIM
240	31.63	0.882	25.62	0.762
288	31.62	0.882	25.62	0.764
336	31.61	0.880	25.59	0.761
384	**31.67**	**0.881**	**25.62**	**0.762**
432	31.64	0.880	25.60	0.760
480	31.58	0.880	25.60	0.760

Effect of candidate set for bit offset We conducted an ablation study to analyze how the range of candidate bit offsets for θ_k used to adjust the bit precision of each layer k affected quantization performance. Here, θ_k denotes a discrete set of candidate values for the bit offset (e.g., $\{-1, 0, 1\}$), which defines how much the bit-width of layer k can be increased or decreased around the base bit-width b_{base}. This experiment aimed to evaluate the impact of the size and span of the candidate values of θ_k on overall performance and stability in a framework where the optimal bit-width for each layer k was selected by applying $\pm\theta_k$ around b_{base}. Table 8 presents the quantitative comparison of PSNR and SSIM results on the Set5 [3] and Urban100 [9] datasets for various candidate values of θ_k, applied to the EDSR model with 4-bit quantization. Figure 5 visualizes the final distribution of weight and activation bit-width per layer when the proposed method was applied to the EDSR [17] ($\times 4$) model. The results showed that an excessively wide candidate set for θ_k tended to increase instability in bit selection, which slightly degraded overall performance. In particular, under a 4-bit base setting, extreme downward adjustments such as $\theta_k = -3$ bits could negatively affect performance due to representational loss. However, this effect may vary depending on the base bit-width or hardware characteristics, so the candidate values for θ_k should be chosen according to the specific deployment context. In our main experiments, we used $\{-1, 0, 1\}$ as the candidate values for θ_k, considering the trade-off between performance and stability.

Loss weighting factor We investigated the influence of the feature-alignment loss weight λ_{feat} on the performance of 4-bit quantized EDSR ($\times 4$). The weighting factor λ_{feat} is introduced in Eq. (1) of the main text, where it balances the contribution of the feature-alignment loss term L_{feat} relative to the reconstruction loss L_{rec} within the overall optimization objective. Since excessively large or small weighting may disrupt stable convergence during quantization, it is important to assess the impact of this hyperparameter. Table 9 reports PSNR and SSIM results on the Set5

Table 11. Ablation study on the effect of **batch size during fine-tuning** quantization bounds for 4-bit EDSR (×4).

Batch size	Set5 (×4)		Urban100 (×4)	
	PSNR	SSIM	PSNR	SSIM
2	**31.67**	**0.881**	**25.62**	**0.762**
4	31.67	0.881	25.62	0.762
8	31.57	0.881	25.61	0.763
16	31.66	0.881	25.62	0.762

[3] and Urban100 [9] datasets when λ_{feat} was varied from 0 to 25. Across all tested values, the performance remained consistently stable, indicating that the influence of λ_{feat} is minimal. Among them, $\lambda_{feat} = 10$ yielded the most stable performance across both datasets, and was therefore adopted for all experiments in the main manuscript. These results verify that the proposed framework is robust to the choice of λ_{feat} and does not require sensitive tuning of this hyperparameter.

Calibrating patch size We further evaluated the effect of calibration patch size on the performance of 4-bit quantized EDSR (×4). Since calibration data are used to estimate activation and weight ranges for quantization, the spatial resolution of input patches may influence the stability of range estimation, especially when only a small calibration set is available. As shown in Tab. 10, PSNR and SSIM results on the Set5 [3] and Urban100 [9] datasets remained highly consistent when the patch size varied from 240 to 480. Among all tested settings, a patch size of 384 provided the most stable performance and was therefore chosen as the default configuration in the main experiments. These results demonstrate that the proposed framework is robust to calibration patch size and does not require precise tuning of spatial resolution during the calibration stage.

Fine-tuning batch size We also analyzed the impact of batch size during the fine-tuning stage of quantization bound adjustment for 4-bit EDSR (×4). Since this step relies on a small calibration set, excessively large or small batch sizes may influence convergence behavior and stability. As reported in Table 11, PSNR and SSIM results on the Set5 [3] and Urban100 [9] datasets exhibited minimal variation across batch sizes ranging from 2 to 16, with batch sizes of 2 and 4 showing nearly identical performance. However, batch size 2 provides the advantage of lower memory consumption and lighter computational overhead during fine-tuning, making it a practical choice. Therefore, batch size 2 was adopted as the default configuration in subsequent experiments. These results confirm that the proposed framework is robust to batch size selection and does not require sensitive hyperparameter tuning.

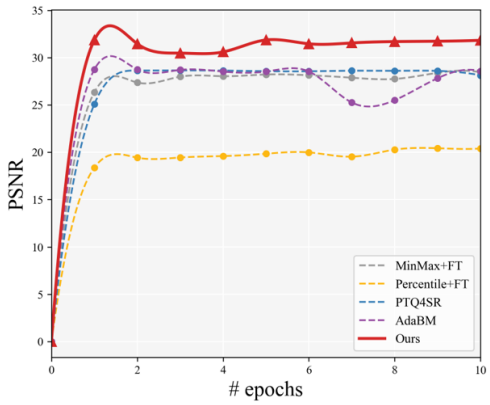

Figure 6. Comparison of fine-tuning performance on Set5 (×4) in terms of PSNR. Our method achieves higher reconstruction quality and **faster convergence** than prior PTQ methods.

Stability and convergence analysis Figure 6 shows the PSNR convergence trend by epoch when the RDN [25] ×4 model was quantized to 4-bit and fine-tuned. The proposed method converged rapidly within just 2 epochs while maintaining stable performance, whereas existing methods either converged more slowly or exhibited unstable performance fluctuations during fine-tuning. In particular, AdaBM [8] employed a strategy of dynamically adjusting the bit-width of each layer throughout the fine-tuning process. However, this approach often caused optimization instability due to inconsistencies between the bit-widths and the corresponding quantization ranges. When the bit-width changed during training, the quantization range was not updated immediately, resulting in the same activation values being quantized with different scales across epochs. This led to unstable oscillations in quantization error and, ultimately, degraded reconstruction performance. In contrast, the proposed method fixed the bit-widths before fine-tuning and pre-adjusted the quantization ranges accordingly. As a result, it was able to quickly recover performance with only brief training, achieving both high convergence stability and efficiency. While other baseline methods showed gradual improvements in performance over fine-tuning epochs, there was a clear gap in both initial convergence speed and final performance compared to the proposed method.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017. 7

[2] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European confer-*

ence on computer vision (ECCV), pages 252–268, 2018. 1, 6

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 6, 7, 10, 11

[4] Arun Chauhan, Utsav Tiwari, et al. Post training mixed precision quantization of neural networks using first-order information. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1343–1352, 2023. 3

[5] Dahun Choi and Hyun Kim. Gradq-vit: Robust and efficient gradient quantization for vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 16019–16027, 2025. 3

[6] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3009–3018. IEEE, 2019. 1

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[8] Cheeun Hong and Kyoung Mu Lee. Adabm: On-the-fly adaptive bit mapping for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2641–2650, 2024. 4, 5, 6, 7, 11

[9] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015. 4, 5, 6, 7, 10, 11

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 1

[11] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018. 1, 2, 4, 6, 7

[12] Souvik Kundu, Shikai Wang, Qirui Sun, Peter A Beerel, and Massoud Pedram. Bmpq: bit-gradient sensitivity-driven mixed-precision quantization of dnns from scratch. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 588–591. IEEE, 2022. 3

[13] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, 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*, pages 4681–4690, 2017. 1, 6

[14] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2810–2819, 2019. 4, 6, 7

[15] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*, 2021. 7

[16] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 1

[17] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 1, 2, 4, 5, 6, 7, 10

[18] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, pages 416–423. IEEE, 2001. 5

[19] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia tools and applications*, 76(20):21811–21838, 2017. 5, 6

[20] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International conference on machine learning*, pages 7197–7206. PMLR, 2020. 7

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 1

[22] Zhijun Tu, Jie Hu, Hanting Chen, and Yunhe Wang. Toward accurate post-training quantization for image super resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5856–5865, 2023. 4, 6, 7

[23] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization. *arXiv preprint arXiv:2203.05740*, 2022. 7

[24] Ke Xu, Zhongcheng Li, Shanshan Wang, and Xingyi Zhang. Ptmq: Post-training multi-bit quantization of neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 16193–16201, 2024. 7

[25] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018. 1, 4, 5, 6, 7, 11