Supplemental Materials - Efficient Visual Computing with Camera RAW Snapshots

Zhihao Li, Ming Lu, Xu Zhang, Xin Feng, M. Salman Asif, and Zhan Ma

***Abstract*—In this supplementary material, we provide additional information to further evidence the generalization of the proposed ρ-*Vision* for the support of extra functionalities. Specifically, we first provide details of our *Unpaired CycleR2R* in Sec. S.I. Then we give proofs of some equations in Sec. S.II. In addition, we demonstrate the potential of run segmentation in the RAW domain directly in Sec. S.III. At last, we show more compression result figures in Sec. S.IV.**

***Index Terms*—Camera RAW, RAW-domain Object Detection, RAW Image Compression**

S.I. Details of the *Unpaired CycleR2R*

A. Architecture of Basic Neural Network

Table I details the architecture of the basic neural network $E(\cdot)$ used in *Unpaired CycleR2R*. This basic network $E(\cdot)$ consists of five layers in total and is used for IEM (Illumination Estimation Module), AWB (Auto White Balance), BA (Brightness Adjustment), and CC (Color Correction). The first layer applies the 5×5 convolution with 32 channels, and the subsequent two layers use 3×3 convolutions and 64 channels. The final two layers use simple linear layers instead.

The example of "Conv: k5c32s2" stands for a convolutional layer having convolutions with spatial kernel size at 5×5 (k5), 32 channels (c32), and a stride of two based spatial downsampling (s2) at both dimensions. The same convention is applied to the linear layer (Linear) and average pooling layer (Avg Pool). "Leaky RELU" [*S1*] is used as the activation, and "Mean" stands for the average operator in the spatial domain for each channel. Considering the output channel of $E(\cdot)$ is specific for different purposes across aforementioned modular components, we mark it using a predefined variable $\mathrm{C_{out}}$.

B. Architectures of Discriminators

As in the main paper, D_{color} and D_{bright} are applied to measure the similarity between generated and real images. D_{color} stacks five convolutional layers with Leaky ReLU [*S1*] and D_{bright} uses five linear layers instead to process 1D grayscale histogram. Details of kernel size, channels, and strides are listed in Tabel. I.

C. Gamma Correction Standard

Gamma correction matches the non-linear characteristics of a display device or human perception [*S2*]. We adopt the correction function recommended in ITU-R BT. 709 standard [*S3*], noted as f_g, which is widely used in commodity ISPs today [*S4*].

$$\begin{aligned}\boldsymbol{y} &= f_g \circ \boldsymbol{x}_{cc} \\ &= \begin{cases} 12.92 \cdot \boldsymbol{x}_{cc}, & \boldsymbol{x}_{cc} \le 0.00304, \\ 1.055 \cdot \boldsymbol{x}_{cc}^{1/2.4} - 0.055, & \boldsymbol{x}_{cc} > 0.00304. \end{cases}\end{aligned} \tag{S1}$$

Correspondingly, the inverse function g_g is:

$$\begin{aligned}\boldsymbol{x}_{cc} &= g_g \circ \boldsymbol{x}_g \\ &= \begin{cases} \dfrac{\boldsymbol{y}}{12.92}, & \boldsymbol{y} \le 0.04045, \\ \left(\dfrac{\boldsymbol{y} + 0.055}{1.055}\right)^{2.4}, & \boldsymbol{y} > 0.04045. \end{cases}\end{aligned} \tag{S2}$$

S.II. Details of *Distribution Analysis of RAW images*

A. The proof of the equation (28)

We start from the loss function $\mathcal{L}$:

$$\mathcal{L} = \frac{1}{H \times W} \left(\boldsymbol{w} * (\boldsymbol{P} - 0.5) + \mathbf{b} - \hat{\mathbf{y}}\right)^2, \tag{S3}$$

where $\boldsymbol{w} \in \mathbb{R}^{S \times S}$ is the convolution kernel with kernel size S.

Then the partial derivative of $w \in \boldsymbol{w}$ could be formulated as:

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{1}{H \times W} \sum_{j=0}^{H} \sum_{i=0}^{W} 2 \left(\mathbf{y}_{ij} - \hat{\mathbf{y}}\right) \left(\mathbf{x}_{ij+mn} - 0.5\right) \tag{S4}$$

where $\mathbf{x}_{ij+mn} \in \boldsymbol{P}$ and mn is the shift position of w according to the kernel center of $\boldsymbol{w}$. $\mathbf{y}_{ij}$ is the convolution output at position ij. To calculate $\mathbf{y}_{ij}$, we define $\boldsymbol{x}_{ij}^{w}$ as a window of $\boldsymbol{P}$ with the same size of $\boldsymbol{w}$ located at ij. Considering the similarity among adjacent pixels, for a neighborhood pixel of $\mathbf{x}_{ij}$, *i.e.*, $\mathbf{x}_{neibor} \in \boldsymbol{x}_{ij}^{w}$, we have $\mathbf{x}_{neibor} = \mathbf{x}_{ij} + \delta$, where

TABLE I: Network settings of *Unpaired CycleR2R*.

Basic Encoder $E(\cdot)$	Discriminator D_{color}	Discriminator D_{bright}
Conv: k5c32s2	Conv: k4c64s2	Linear: c1024
Leaky RELU	Leaky RELU	Leaky RELU
Avg Pool: s2	Conv: k4c128s2	Linear: c1024
Conv: k3c64s2	Leaky RELU	Leaky RELU
Leaky RELU	Conv: k4c256s2	Linear: c256
Avg Pool: s2	Leaky RELU	Leaky RELU
Conv: k3c64s1	Conv: k4c512s2	Linear: c256
Leaky RELU	Leaky RELU	Leaky RELU
Mean	Conv: k4c1s2	Linear: c1
Linear: c256	Mean	-
Linear: $\mathrm{cC_{out}}$	-	-

δ follows a Gaussian distribution with zero mean. Thus, $\mathbf{y}_{ij}$ could be expanded as:

$$\mathbf{y}_{ij} = (\mathbf{x}_{ij} - 0.5)\sum w + + \sum w\delta. \tag{S5}$$

For simplify, we use $\tilde{\boldsymbol{x}}$ and $\tilde{\mu}$ to replace $\boldsymbol{x} - 0.5$ and $\mu - 0.5$, respectively. Besides, we set $A = \sum w,\ B = + \sum w\delta,\ C = 2A,\ D = 2\,(B - \hat{\mathbf{y}})$. Having $\mathbf{y}_{ij} = A\tilde{\mathbf{x}} + B$, the $\frac{\partial \mathcal{L}}{\partial w}$ will be:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w} &= 2\mathbb{E}\left[(\mathbf{y} - \hat{\mathbf{y}})\,(\mathbf{x} - 0.5)\right] \\
&= 2\mathbb{E}\left[(A\tilde{\mathbf{x}} + B - \hat{\mathbf{y}})\,\tilde{\mathbf{x}}\right] \\
&= 2\mathbb{E}\left[A\tilde{\mathbf{x}}^2 + (B - \hat{\mathbf{y}})\tilde{\mathbf{x}}\right] \\
&= 2\mathbb{E}\left[A\right]\mathbb{E}\left[\tilde{\mathbf{x}}^2\right] + 2(\mathbb{E}\left[B\right] - \hat{\mathbf{y}})\mathbb{E}\left[\tilde{\mathbf{x}}\right] \\
&= 2A(\tilde{\mu}^2 - \sigma^2) + 2(b - \hat{\mathbf{y}})\tilde{\mu} \\
&= C(\tilde{\mu}^2 - \sigma^2) + D\tilde{\mu}.
\end{aligned} \tag{S6}$$

Since μ and σ are independent and we only concern with the impact of $p\,(\mu)$, we set Var $\left[\sigma^2\right]$ to a constant. Then the variance could be expanded as:

$$\begin{aligned}
\text{Var}\left[\frac{\partial \mathcal{L}}{\partial w}\right] &= \text{Var}\left[C\tilde{\mu}^2 + D\tilde{\mu}\right] + \text{Var}\left[C\sigma^2\right] \\
&= \mathbb{E}\left[\left(C\tilde{\mu}^2 + D\tilde{\mu}\right)^2\right] - \mathbb{E}\left[C\tilde{\mu}^2 + D\tilde{\mu}\right]^2 \\
&\quad + \text{const} \\
&= \mathbb{E}\left[C^2\tilde{\mu}^4\right] + \cancel{\mathbb{E}\left[CD\tilde{\mu}^3\right]} + \mathbb{E}\left[D^2\tilde{\mu}^2\right] \\
&\quad - \left(\mathbb{E}\left[C\tilde{\mu}^2\right] + \cancel{\mathbb{E}\left[D\tilde{\mu}\right]}\right)^2 + \text{const} \\
&= \mathbb{E}\left[\left(C\tilde{\mu}^2\right)^2\right] - \left(\mathbb{E}\left[C\tilde{\mu}^2\right]\right)^2 \\
&\quad + \mathbb{E}\left[\left(D\tilde{\mu}\right)^2\right] - \left(\mathbb{E}\left[D\tilde{\mu}\right]\right)^2 + \text{const} \\
&= \text{Var}\left[C\tilde{\mu}^2\right] + \text{Var}\left[D\tilde{\mu}\right] + \text{const} \\
&= C^2\text{Var}\left[\tilde{\mu}^2\right] + D^2\text{Var}\left[\tilde{\mu}\right] + \text{const}.
\end{aligned} \tag{S7}$$

B. The proof of the equation (29)

Given the μ following the distribution in (25), the Var $[\tilde{\mu}]$ could be written as:

$$\begin{aligned}
\text{Var}\left[\tilde{\mu}\right] &= \text{Var}\left[\mu - 0.5\right] = \text{Var}\left[\mu\right] \\
&= \int_0^1 \left[\mu - \mathbb{E}\left(\mu\right)\right]^2 p\left(\mu\right) \mathrm{d}\mu \\
&= \int_0^1 \left(\mu - 0.5\right)^2 \left(k\mu^2 - k\mu + \frac{k}{6} + 1\right)\mathrm{d}\mu \\
&= F(\mu = 1) - F(\mu = 0) \\
&= \left(\frac{1}{21} - \frac{k}{720}\right) - \left(-\frac{k}{144} - \frac{1}{24}\right) \\
&= \frac{k}{180} + \frac{1}{12},
\end{aligned} \tag{S8}$$

where $F\left(\mu\right) = k\left(\frac{\mu^5}{5} - \frac{\mu^4}{2} + \frac{\mu^3}{12} - \frac{\mu^2}{8}\right) + \frac{k}{18}\left(\mu - \frac{1}{2}\right)^3$.

Thus, Var $\left[\frac{\partial \mathcal{L}}{\partial w}\right]$ will be:

$$\begin{aligned}
\text{Var}\left[\frac{\partial \mathcal{L}}{\partial w}\right] &\approx D^2\text{Var}\left[\tilde{\mu}\right] + \text{const} \\
&= D^2\left(\frac{k}{180} + \frac{1}{12}\right) + \text{const} \\
&= D^2\frac{k}{180} + \text{const}.
\end{aligned} \tag{S9}$$

S.III. RAW-domain Segmentation

In the main text of this paper, the detection task is successfully executed in the RAW domain with superior performance to that using the same RGB-domain model. Here we explore the feasibility of RAW-domain segmentation. Similar to the discussions in Sec. 5.2 and 6.2, we first demonstrate that the segmentation model trained with simRAW images can directly infer the segmentation cues upon the real RAW images. Second, a few-shot finetuning simRAW-pretrained segmentation model using limited labeled real RAW images further improves its performance and shows consistent gains to the model trained from scratch. At last, ablation studies shows that gamma correction is also vital for segmentation task in the RAW domain.

A. Datasets

Cityscapes [S5] is a large-scale dataset recorded in different urban streets in Europe, which contains 5,000 frames with high-quality pixel-level segmentation annotations. Considering the different traffic signs in China where the *MultiRAW* is captured, we use a communal subset including road, building, fence, traffic light, sky, person, car, truck, and bus for evaluation. Following the setup in Sec. 5.2 of the main paper, we convert the RGB samples, a.k.a RGB_c, into simRAW image set simRAW_c to train/refine RAW-domain segmentation model.

B. Training Details

We use the famous HRNetv2 [S6] as our segmentation network. All segmentation models are optimized by a SGD optimizer with 0.9 momentum, 5×10^{-4} weight decay and initial learning rate of 10^{-2} dropped into 10^{-4} linearly. The batch size is set as 8 and inputs are randomly cropped into 512×1024 with random flip augmentation. The experiments are conducted using an Nvidia 3090Ti GPU.

C. Comparative Studies of RAW-domain Segmentation

Table II and Fig. 1 compares our *Unpaired CycleR2R* and other methods using invISP approach [S9, S10, S11, S12] and domain-adaptation (DA) solution [S7, S8]. It can be seen that *Unpaired CycleR2R* outperforms the state-of-the-art CycleISP by a significant margin of 7 mIoU and improves the IoU across all classes of objects. More gains are presented against other approaches.

Our model also surpasses the RGB Baseline on mIoU. Note that this RGB Baseline is prevalent in real-world applications. Such a convincing performance suggests the potential for RAW-domain segmentation. We also observe the lower IoU for some specific classes of objects between our method and the RGB Baseline. This is probably due to the optimization strategy used for maximizing the overall performance but not balancing each class. This is an interesting topic for future study.

Apparently, inputting real RAW images to the RGB-domain segmentation model directly for task execution is a failure, as exemplified in the Naive Baseline, e.g., mIoU of 11.1 versus the mIoU of 47.5 in the RGB Baseline, which is due to the

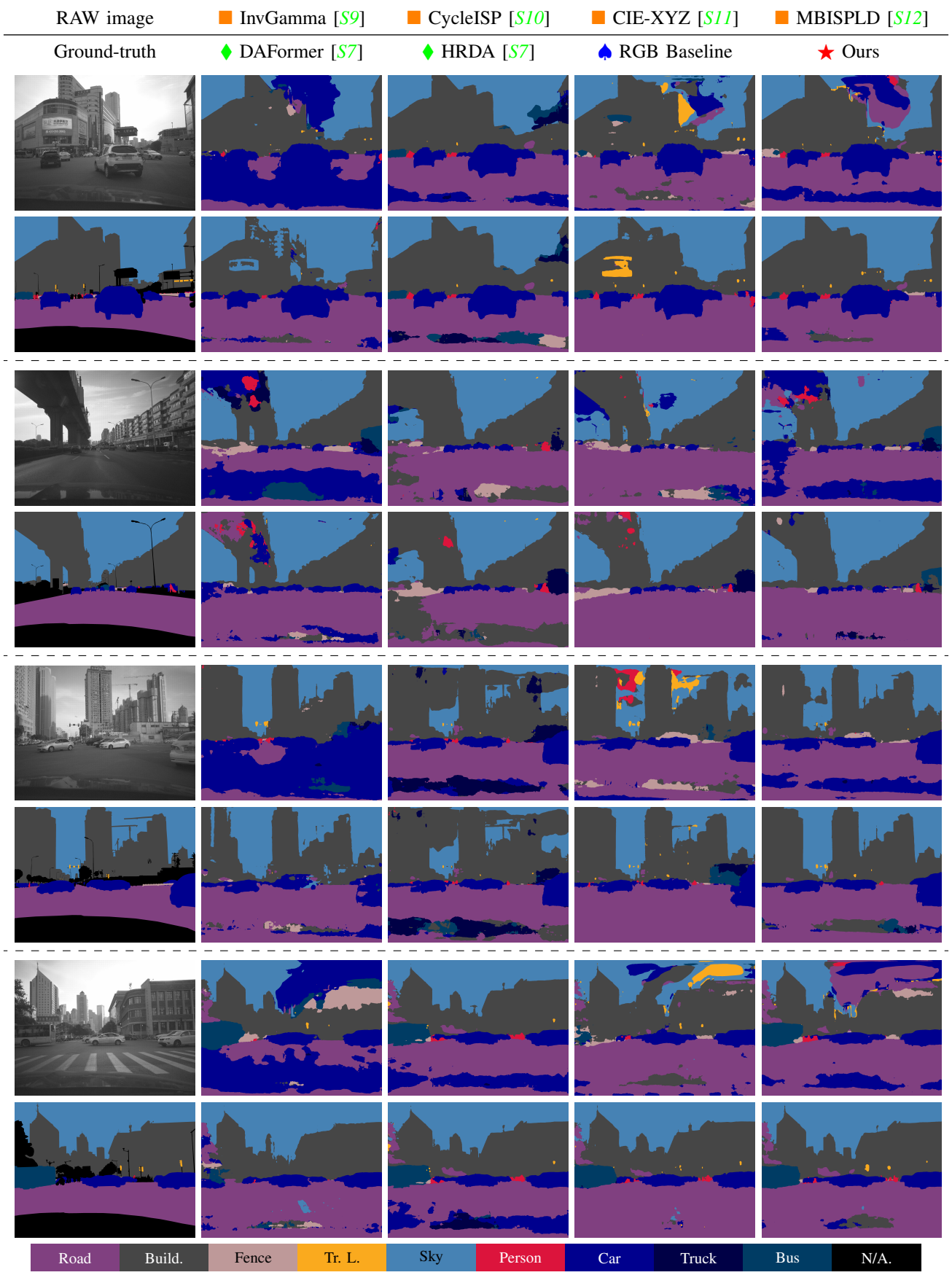

Fig. S1: **Qualitative Visualization of Pretrained RAW Segmentation Model.** Example predictions show better recognition of buildings, sky, and traffic lights by our *Unpaired CycleR2R* on Cityscapes RGB → iPhone RAW. Gamma correction and brightness adjustment have been applied to RAW images for a better view.

TABLE II: mIoU (Mean Intersection over Union) of Segmentation on the testing set of iPhone RAW images. RGB-domain segmentation model is trained using original RGB images in *Cityscapes* [S5] (e.g., RGB_c); Various simRAW datasets associated with RGB_c are generated using different methods which are marked as $simRAW_c$ to train RAW-domain segmentation model. The testing RAW images in iPhone RAW RAW_i and their paired RGB images in RGB_i converted using built-in iPhone ISP are tested accordingly. HRNetv2 [S6] is used as the base segmentation model. ♠ Baselines, ♦ Domain Adaptation Solutions, ■ invISP Methods, ★ Ours.

Method	invISP	Segmentor		Road	Build.	Fence	Tr. L.	Sky	Person	Car	Truck	Bus	mIoU
	Train	Train	Test										
♠ Naive Baseline	-	RGB_c	RAW_i	0.3	21.6	14.8	5.7	20.7	0.4	30.0	0.4	6.2	11.1
♠ RGB Baseline	-	RGB_c	RGB_i	**89.6**	65.1	35.6	20.7	**96.1**	11.1	62.9	**21.5**	25.3	47.5
♦ DAFormer (CVPR'22) [S7]	-	RGB_c, RAW_i	RAW_i	75.8	49.5	15.2	1.5	90.0	5.3	58.3	0.2	6.3	32.9
♦ HRDA (ECCV'22) [S8]	-	RGB_c, RAW_i	RAW_i	73.8	69.1	38.5	12.3	80.6	15.0	51.2	16.2	20.9	42.0
■ InvGamma (ICIP'19) [S9]	RGB_i, RAW_i	$simRAW_c$	RAW_i	47.5	55.7	31.2	8.3	90.0	7.3	23.9	11.2	17.6	32.5
■ CycleISP (CVPR'20) [S10]	RGB_i, RAW_i	$simRAW_c$	RAW_i	84.8	63.9	35.0	18.0	86.3	9.7	55.7	18.0	20.6	43.6
■ CIE-XYZ Net (TPAMI'21) [S11]	RGB_i, RAW_i	$simRAW_c$	RAW_i	78.7	64.4	36.7	3.0	84.2	5.4	48.6	2.3	15.4	37.6
■ MBISPLD (AAAI'22) [S12]	RGB_i, RAW_i	$simRAW_c$	RAW_i	72.5	60.8	39.4	7.3	78.6	13.3	41.0	17.7	20.8	39.0
★ **Unpaired CycleR2R**	RGB_c, RAW_i	$simRAW_c$	RAW_i	88.9	**70.5**	**40.9**	**24.7**	95.5	**21.4**	**64.3**	19.1	**30.0**	**50.6**

Build. ← Building; **Tr. L.** ← Traffic Light.

large discrepancy between the RGB-domain and RAW-domain models.

Implementation Friendliness. As aforementioned in Sec. 5.2, our method could generate simRAW images to train task-dependent models. However, DA-based approaches [S13, S14] designed for object detection tasks could not be applied to segmentation tasks. And DA-based segmentation methods [S7, S8] could not support the detection task either.

D. Comparative Studies of Few-Shot Finetuning

The performance of the simRAW-pretrained segmentation model could be further boosted by feeding more real labeled RAW images. We further finetune our segmentation model using our *MultiRAW* dataset (iPhone XSmax) with all classes. As depicted in Fig. 2, the segmentation accuracy is improved and consistently outperforms the scratch model which is initialized randomly and then trained using the same labeled real RAW images.

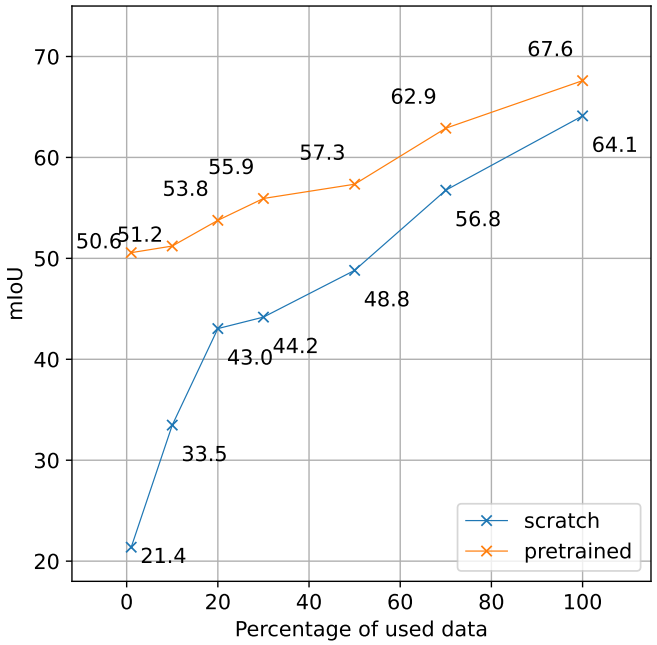

Fig. S2: **Few-shot finetuning using limited camera RAWs.** The simRAW-pretrained HRNetv2 [S6] is obtained by using samples in $simRAW_c$ generated by our Unpaired CycleR2R, which is then finetuned using limited camera RAW images; and the "scratch" model is randomly initialized and then trained using the same number of labeled real RAW images.

S.IV. Extra Quantitative Visualization

We also offer more qualitative visualizations of our lossy RIC at low Bits-rate and high Bits-rate in FIg. 3 and Fig. 4 respectively. Similar to the results in the main content of this work, we can clearly observe the subjective improvements of the proposed lossy RIC compared to the HEVC and VVC. Especially for the traffic light and car information, our lossy RIC provides sharper and less noisy reconstructions that are closer to the ground truth samples. Also, we give visualizations of progressive decoding using our lossless RIC within various cameras in Fig. 5-7. Our lossless RIC could provide low-resolution previews for different cameras (iPhone XSmax, Huawei P30pro, and asi 294mcpro) and different scenes (both daylight and nighttime), which is helpful for professional photography and network transmission.

References

[S1] J. T. Barron and Y.-T. Tsai, "Fast fourier color constancy," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 886–894. 1

[S2] H. Farid, "Blind inverse gamma correction," *IEEE*

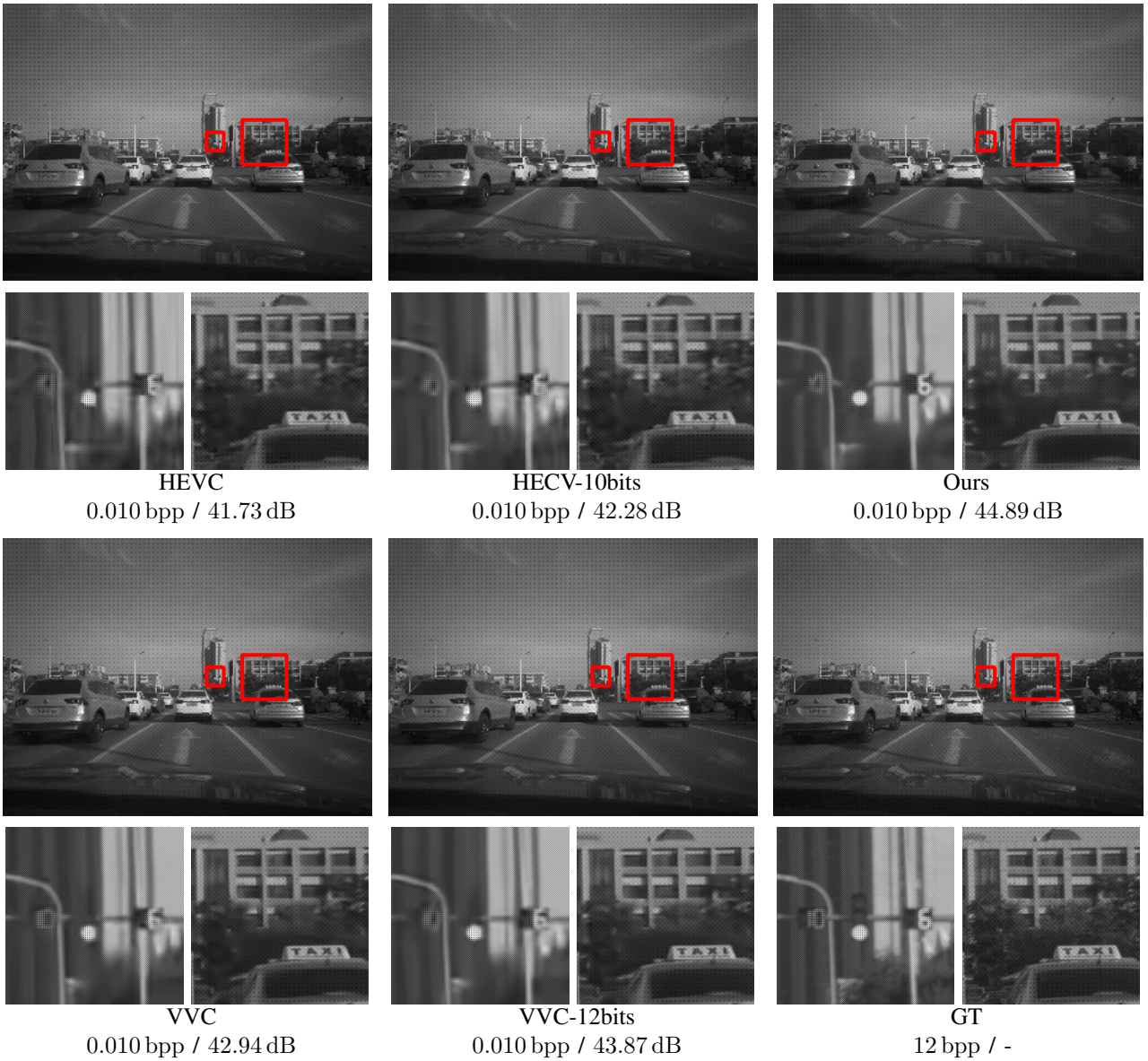

Fig. S3: **Qualitative Visualization of Lossy RIC at Low Bits-rate.** Reconstructions and close-ups of the HEVC, VVC, and our method. Corresponding bpp and PSNR are marked. Gamma correction and brightness adjustment have been applied for a better view. *Zoom for better details.*

transactions on image processing, vol. 10, no. 10, pp. 1428–1433, 2001. 1

[*S3*] M. Stokes, "A standard default color space for the internet-srgb," *http://www. color. org/contrib/sRGB. html*, 1996. 1

[*S4*] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, "Adaptive logarithmic mapping for displaying high contrast scenes," in *Computer graphics forum*, vol. 22, no. 3. Wiley Online Library, 2003, pp. 419–426. 1

[*S5*] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4

[*S6*] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020. 2, 4

[*S7*] L. Hoyer, D. Dai, and L. Van Gool, "Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9924–9935. 2, 3, 4

[*S8*] ——, "Hrda: Context-aware high-resolution domain-adaptive semantic segmentation," *arXiv preprint arXiv:2204.13132*, 2022. 2, 4

[*S9*] S. Koskinen, D. Yang, and J.-K. Kämäräinen, "Reverse

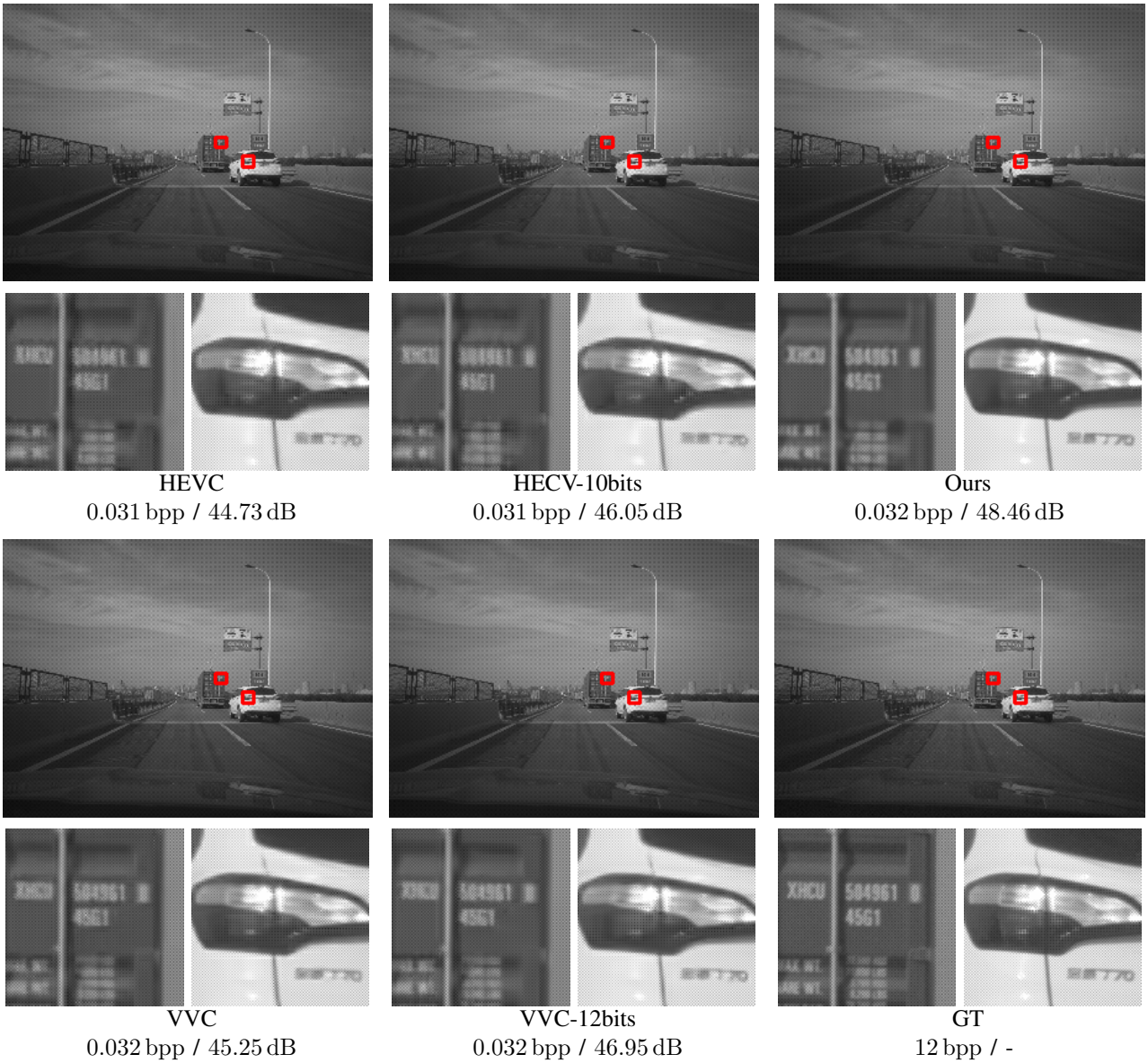

Fig. S4: **Qualitative Visualization of Lossy RIC at High Bits-rate.** Reconstructions and close-ups of the HEVC, VVC, and our method. Corresponding bpp and PSNR are marked. Gamma correction and brightness adjustment have been applied for a better view. *Zoom for better details.*

imaging pipeline for raw rgb image augmentation," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 2896–2900. 2, 3, 4

[*S10*] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Cycleisp: Real image restoration via improved data synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2696–2705. 2, 3, 4

[*S11*] M. Afifi, A. Abdelhamed, A. Abuolaim, A. Punnappurath, and M. S. Brown, "CIE XYZ Net: Unprocessing images for low-level computer vision tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2, 3, 4

[*S12*] M. V. Conde, S. McDonagh, M. Maggioni, A. Leonardis, and E. Pérez-Pellitero, "Model-based image signal processors via learnable dictionaries," in *AAAI*, 2022. 2, 3, 4

[*S13*] M. Hnewa and H. Radha, "Multiscale domain adaptive yolo for cross-domain object detection," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 3323–3327. 4

[*S14*] Y.-J. Li, X. Dai, C.-Y. Ma, Y.-C. Liu, K. Chen, B. Wu, Z. He, K. Kitani, and P. Vajda, "Cross-domain adaptive teacher for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7581–7590. 4

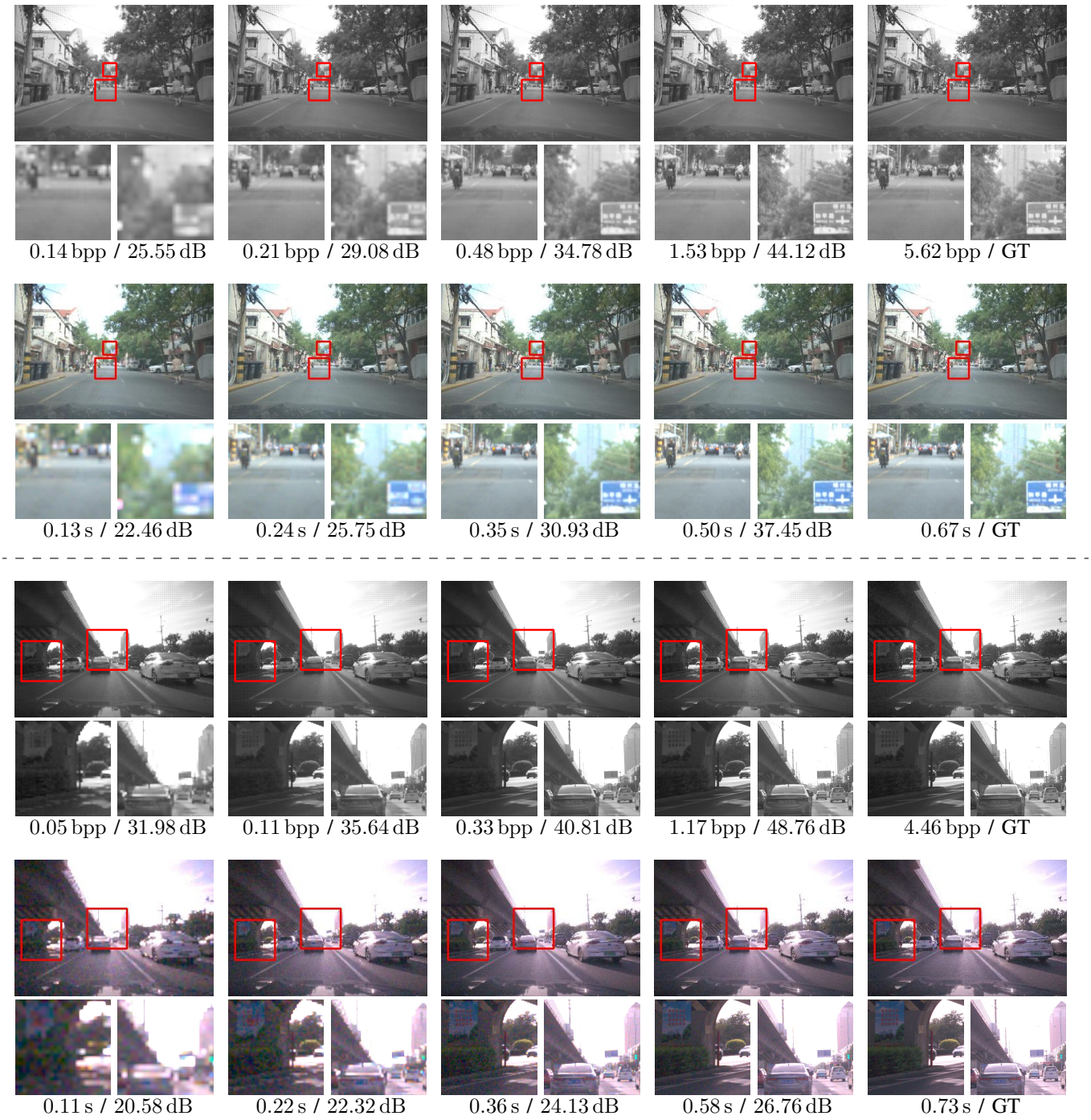

Fig. S5: **Qualitative Visualization of Lossless RIC Progressive Decoding (iPhone XSmax).** The gradual reconstruction of RAW images and their corresponding RGB images converted by an in-camera ISP. Bits per pixel (bpp) / PSNR (dB) is shown under RAW images. Decoding latency (s) / PSNR (dB) is also listed below RGB images. PSNR is derived against the GT (ground truth). Gamma correction and brightness adjustment have been applied for a better view. *Zoom for more details.*

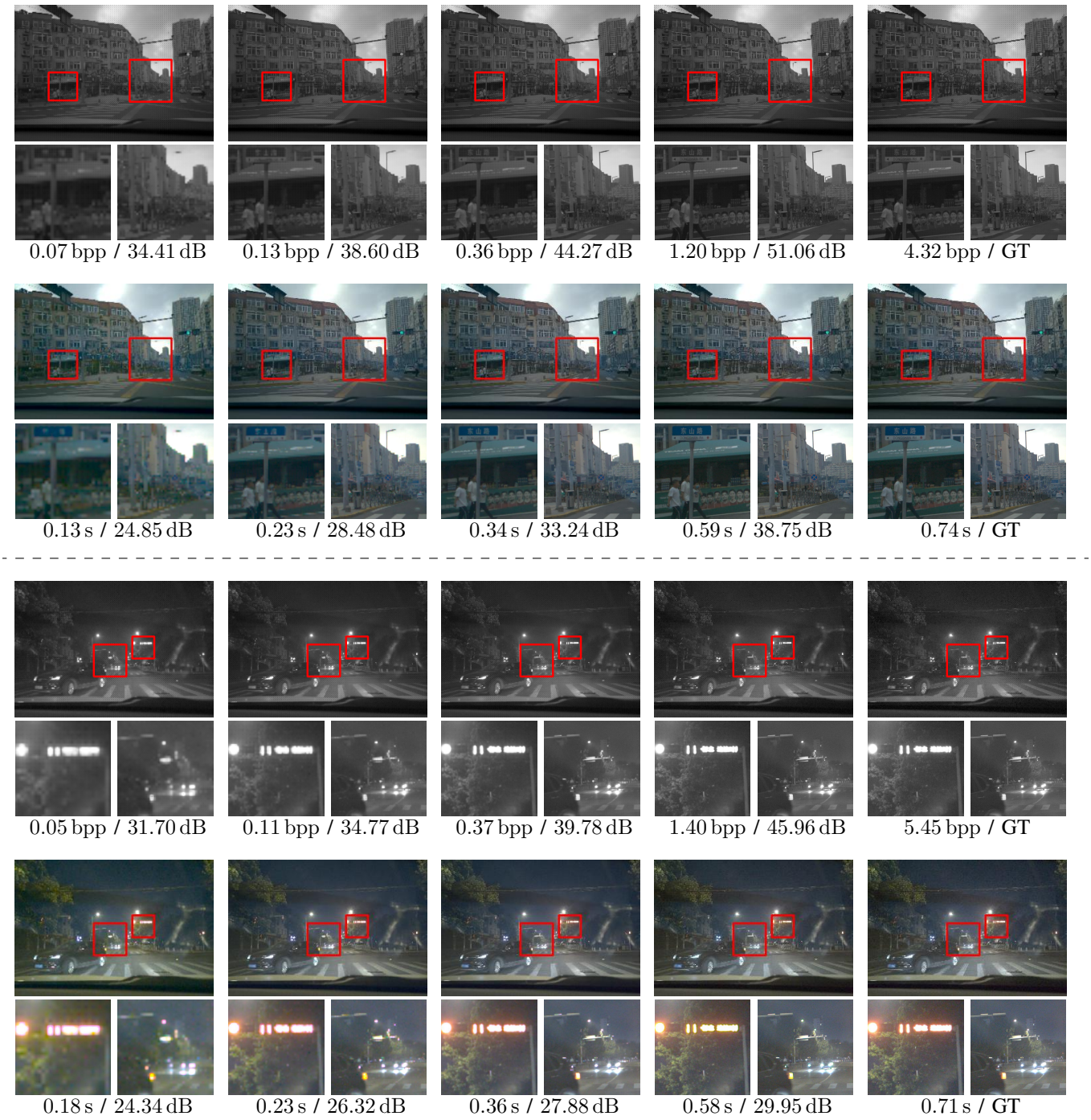

Fig. S6: **Qualitative Visualization of Lossless RIC Progressive Decoding (Huawei P30pro).** The gradual reconstruction of RAW images and their corresponding RGB images converted by an in-camera ISP. Bits per pixel (bpp) / PSNR (dB) is shown under RAW images. Decoding latency (s) / PSNR (dB) is also listed below RGB images. PSNR is derived against the GT (ground truth). Gamma correction and brightness adjustment have been applied for a better view. *Zoom for more details.*

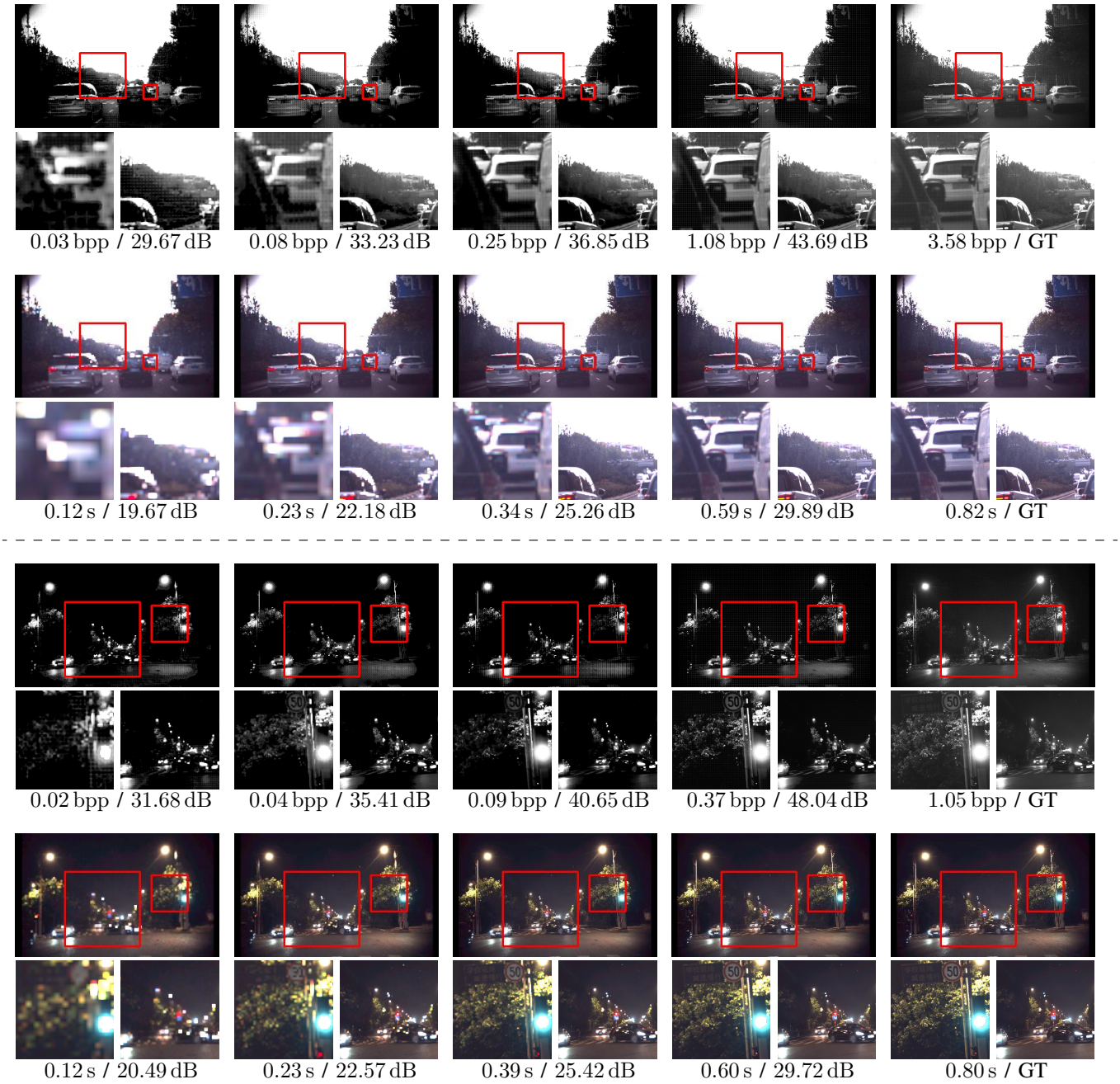

Fig. S7: **Qualitative Visualization of Lossless RIC Progressive Decoding (asi 294mcpro).** The gradual reconstruction of RAW images and their corresponding RGB images converted by an in-camera ISP. Bits per pixel (bpp) / PSNR (dB) is shown under RAW images. Decoding latency (s) / PSNR (dB) is also listed below RGB images. PSNR is derived against the GT (ground truth). Gamma correction and brightness adjustment have been applied for a better view. *Zoom for more details.*