CoDISP: Exploring Compressed Domain Camera ISP with RGB-guided Encoder

Molin Zhang[1*] Soumendu Majee[2] Chengyu Wang[2] Seok-Jun Lee[2]
Hamid Sheikh[2]
[1]Massachusetts Institute of Technology,[2] Samsung Research America
molin@mit.edu,{s.majee,chengyu.wang,seokjun1.lee,hr.sheikh}@samsung.com

Abstract

Most mobile device Image Signal Processing (ISP) pipelines operate directly on RAW image data for all processing tasks. However, the rise of super-high-resolution cameras on mobile devices has led to increased memory demands for multi-frame ISP pipelines. In this work, we introduce a novel ISP pipeline that operates on a learned compressed domain, aiming to conserve memory for downstream ISP modules' inputs. We utilize RGB image compression to define a compressed latent domain, preserving both semantic information and high-frequency details. To facilitate mapping of raw images to the compressed domain, we develop a transfer learning strategy. All downstream processing tasks, including demosaicing, single and multi-frame denoising, and registration, are performed on this compressed latent domain. We demonstrate the effectiveness of our compressed domain ISP pipeline on both public and internal datasets. Remarkably, our pipeline achieves ISP performance similar to non-compression methods while significantly reducing mobile memory requirements.

1. Introduction

The image Signal Processing (ISP) pipeline plays an important role in mobile camera imaging. Recently, deep learning (DL) has demonstrated remarkable success in various computational photography and vision applications due to its outstanding performance and faster computation time. This success spans across both low-level tasks, such as demosaicing [37, 48], denoising [24, 28, 38], and super-resolution [9], as well as high-level tasks, such as tone mapping [32], detection, and pose estimation [43, 49, 50, 52]. Additionally, DL has shown promising capabilities in the Image Signal Processing (ISP) pipeline [16, 25, 34]. The emergence of Neural Processing Units (NPUs) and Graphics Processing Units (GPUs) in mobile platforms has provided the necessary hardware support to deploy DL models on modern mobile phones.

However, the high-resolution camera sensors in modern mobile devices dramatically increase the memory requirement for current ISP algorithms and multi-frame image techniques, creating a major bottleneck for further image quality (IQ) improvements. Smartphone cameras generally use multiple frames for low resolution photos but could be limited to 1 or 2 frame only for high megapixel cameras. Current DL-based ISP pipelines also directly work on full-resolution images which restrict the performance of multi-frame image processing [16, 25, 34].

Image compression techniques are proposed to overcome the limitation of memory storage and data transferring. Conventional image compression techniques like JPEG [46] and BPG [6] rely heavily on hand-crafted compression features and rules. Those features exhibit occasional challenges in effective generalization, particularly within intricate scenarios, leading to an observable decline in overall performance. Learned compression rules with deep learning, especially Convolutional Neural Networks (CNN) have been proven to achieve outperforming compression performance [4, 5, 7, 8, 14] compared to traditional methods. Compressed domain has shown success in various applications including generative models and solving inverse problems [2, 53]. Current compression techniques involve entropy encoding for a higher compression ratio in data transfer. For example, the variational auto-encoder (VAE) framework [19] has been widely adopted for high distortion-rate ratio [35]. To better retrieve the details from the compression-decompression process, window-based attention [10, 36, 45] has also been introduced into the encoder and decoder [47, 54].

In this paper, we introduce a novel approach to ISP pipeline directly operating on a learned compressed domain, aimed at mitigating high memory usage by employing compressed latent variables to reduce input size for ISP modules. To the best of our knowledge, we are the first to propose such an ISP pipeline directly on the compressed domain.

There are two main objectives in this study:

*Work done while interning at Samsung Research America MPI lab.

1. ***Designing a compressed latent manifold*** to enable compression and projection of RAW images onto the manifold.
2. ***Demonstrating the effectiveness of the compressed latent manifold*** in performing downstream ISP tasks.

We utilize RGB image compression techniques to define the latent space. RGB images inherently contain more semantic information and details compared to RAW images. Once a raw image is captured, we employ another encoder specifically designed for raw images, following the same architecture as the RGB encoder, to compress it onto the same compressed latent space defined by RGB compression. We utilize transfer learning to train the raw image encoder by transferring the model parameters to facilitate the mapping between the compressed raw image and compressed RGB image. This approach ensures that both raw and RGB images can be compressed uniformly, sharing the same compressed latent manifold. In addition to compression and demosaicing, we demonstrate the capability of our compressed domain ISP for other downstream ISP tasks such as denoising and registration, where separate deep learning models are employed with compressed inputs. Our experimental results show that our pipeline achieves similar ISP performance while significantly reducing mobile memory requirements. From the results, our pipeline achieves similar ISP performance while reducing the mobile memory requirements by reducing the size of input to the ISP modules. Note that while a conventional ISP pipeline typically includes tasks such as tone mapping and multi-frame blending, this work primarily focuses on establishing a proof-of-concept for an ISP pipeline directly on a compressed domain. The main contributions of this paper are summarized as follows:

- We propose a novel ISP pipeline directly on the compressed domain to save memory for high-resolution images captured by mobile phone cameras.
- We introduce a transfer learning strategy to enhance the training of the raw image encoder, improving its ability to map raw image data to the compressed domain.
- We evaluate and demonstrate the ability of our compressed domain ISP for other downstream tasks with three representatives: demosaicing, denoising and registration.

2. Related Works

2.1. RGB image compression

Traditional RGB image lossy compression algorithms, such as JPEG [33] and JPEG2000 [39], employ hand-crafted compression rules for image compression. Recent deep learning (DL) methods based on Convolutional Neural Networks (CNNs) have demonstrated significant advantages over traditional image compression methods by directly optimizing Shannon's Rate-Distortion (R-D) trade-off. Initial attempts were made using Recurrent Neural Network (RNN) architectures [41, 42] and autoencoder models [4, 40]. Modern efficient image compression frameworks are based on Variational Autoencoder (VAE) architectures. When a distortion metric is specified, VAEs are trained to compress data by minimizing a tight upper bound on the Rate-Distortion (R-D) loss function.

To better model the posterior of compressed latent variables, a hyper-prior was incorporated into the VAE framework to capture the spatial dependence [5]. Side information is generated with additional network modules and used as conditional information for compressed information where the posterior becomes Gaussian distribution [7, 23, 26]. The prior for the side information is modeled as a non-parametric kernel-based probability. The overall optimization function with side information is modeled below,

$$\begin{aligned}\mathcal{L} =& R + \lambda \cdot D \\ =& \mathbb{E}_{x\sim p_x}\left[-\log_2 p_{\hat{y}|\hat{z}}(\hat{y} \mid \hat{z}) - \log_2 p_{\hat{z}}(\hat{z})\right] \\ &+ \lambda \cdot \mathbb{E}_{x\sim p_x}[d(x, \tilde{x})]\end{aligned} \tag{1}$$

where R represents ratio loss, D represents distortion loss, x is the original image, $\tilde{x}$ is the reconstructed image, $\hat{y}$ quantized is the compressed latent variables and $\hat{z}$ is the side information. λ is the hyperparameter controlling the trade-off between bit-rate and distortion.

To further enhance the hyperprior entropy model, autoregressive components are utilized to recover the missing part of the quantization [29]. Generative Adversarial Networks (GANs) [13] have been applied to enhance image quality by introducing additional perceptual control [27, 44]. At the same time, the R-D trade-off becomes a triple trade-off between rate distortion and perception.

2.2. Raw image demosaicing

A raw image from a phone camera typically has only one color channel due to a specific Color Filter Array (CFA) pattern. Demosaicing methods convert these single-channel raw images into full-color RGB images. While various non-deep learning-based demosaicing algorithms exist, many struggle with preserving high-frequency structures, leading to artifacts. Several approaches aim to improve recovery in high-frequency regions [11, 15, 31], but they often suffer from issues like excessive blurring, false colors, and high computational demands.

DL-based demosaicing methods offer superior performance by replacing hand-crafted features with learned demosaicing rules. Emerging techniques combine demosaicing with denoising for joint improvement. For example, Gharbi et al.[12] trained a deep convolutional neural network on a large dataset, achieving leading performance. Additionally, Kokkinos et al.[21] introduced an iterative network that integrates the majorization-minimization algorithm with a residual denoising network.

Among DL-based algorithms, demosaicing tasks typically utilize information from a full-size image, often incorporating residual links in network architectures. In contrast, our approach presents a demosaicing framework in the compressed domain, without residual links and using compressed intermediate variables compared to the input raw image.

3. Proposed compressed domain ISP

Designing a compression method and its corresponding compressed latent manifold for downstream tasks in ISP is a non-trivial task. To achieve the desired performance, we propose a novel compressed domain ISP with an RGB-guided compression encoder, as illustrated in Figure 1.

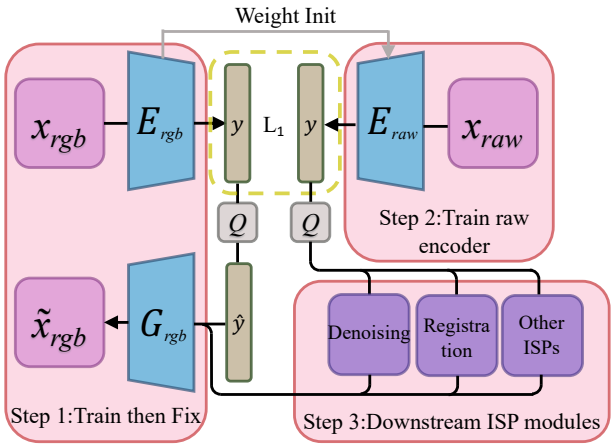

Figure 1. Our proposed compressed domain ISP framework involves three key steps. Firstly, an RGB encoder and decoder are trained using RGB images to define a latent manifold. Secondly, transfer learning is employed to train a raw image encoder with an identical network architecture as the RGB counterpart. Initialized with the weights of the RGB encoder, the raw image encoder is trained to map raw image data onto the established latent manifold using L1 loss optimization. Finally, downstream ISP modules are trained directly on the latent manifold, demonstrated here through denoising and registration modules.

3.1. RGB image compression

VAE-based learned RGB image compression achieves high fidelity reconstruction quality and low data transferring bitstream, but the trade-off between the distortion and the compression ratio tends to over-smooth the regions with high-frequency details when the compression ratio is high as shown in figure 2. Here we directly minimize the distortion cost function so that the compressed latent manifold contains richer details and semantic information, and the reconstructed images have improved image quality. Following the encoder-decoder $(E_{\mathcal{RGB}}(\cdot), G_{\mathcal{RGB}}(\cdot))$ architecture deployed in most autoencoder-based or VAE-based works without side information modules, the loss function with L2 loss is formulated as below:

$$\mathcal{L}_{\mathcal{RGB}} = \mathbb{E}_{x_{rgb}\sim p_x}[\mathcal{L}_2(x_{rgb}, G_{\mathcal{RGB}}(\mathcal{Q}(E_{\mathcal{RGB}}(x_{rgb}))))], \tag{2}$$

where $\mathcal{Q}(\cdot)$ represents the quantization operator. The zeros gradient issue of the quantization can be bypassed in PyTorch with *x=torch.round(x)+x-x.detach()* during the gradient updates.

We define the compressed latent manifold as the output from the encoder $E_{\mathcal{RGB}}(x)$, containing the information required for RGB image reconstruction. The compression ratio is defined as the ratio between the sizes of the input RGB image and the latent variables, which are quantized to 8-bit integers. This fixed compression ratio design facilitates downstream tasks directly on the learned latent manifold.

In conclusion, we prioritize a fixed compression ratio when designing the compressed domain ISP. VAE-based compression methods with R-D loss function often result in overly smoothed reconstruction images when the compression ratio is high, diminishing the effectiveness of the compressed latent manifold for downstream ISP tasks. Additionally, even if entropy encoding is considered, we prefer a constant compressed size for ISP purposes. This approach ensures that we allocate a fixed size, as uncertainty in size allocation would necessitate a larger size regardless. Therefore, we do not utilize entropy loss in our work.

3.2. RGB-guided raw image compression

The inputs to the camera ISP pipeline are raw images acquired by a certain CFA pattern. Building a compressed domain ISP directly using raw images is inferior compared to RGB images due to the training difficulty with high-frequency CFA binary masks, and the latent manifold trained with single-channel inputs contains fewer details and semantic information.

To alleviate the difficulties in training and the build-up of the compressed latent manifold of raw images x_{raw}, we introduce a novel method where the raw image encoder and RGB image encoder share the same compressed latent manifold defined by the trained RGB encoder $E_{\mathcal{RGB}}(\cdot)$ and decoder $G_{\mathcal{RGB}}(\cdot)$. The RGB image encoder guides the training of the raw image encoder $E_{\mathcal{RAW}}(\cdot)$ by enforcing the mapping of the same object or image to the same latent point. The same encoder network architecture is used and initialized with the parameters of the RGB encoder. To ensure the input is compatible with the architecture of the shared encoder, we rearrange the values from the single-channel CFA pattern into the three-channel RGB format and zero-fill the missing values. The loss function for raw image compression is formulated below,

$$\mathcal{L}_{\mathcal{RAW}} = \mathbb{E}_{x_{rgb}\sim p_x}[\mathcal{L}_1(E_{\mathcal{RAW}}(x_{raw}), E_{\mathcal{RGB}}(x_{rgb}))] \tag{3}$$

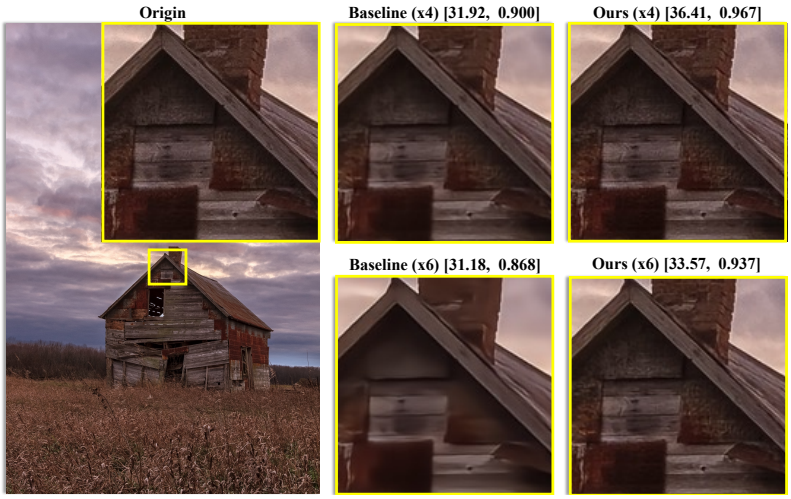

Figure 2. Illustration of RGB image compression with (baseline) and without (ours) entropy encoding loss and auto-regressive module for side information. Evaluation metrics are shown as [PSNR, SSIM]. For both compression at 4 and 6, training RGB image compression without entropy encoding yields better image quality with finer details.

Compared with previous work [17], where an RGB encoder-decoder is also used as a teacher model, the guidance is employed in the training of the RAW domain decoder. Here, we guide the RAW domain encoder to map onto the trained compressed latent manifold. Moreover, while the ISPs in the cited work are performed on the full resolution RGB domain, our approach performs ISPs on the compressed latent manifold.

We choose the L1 loss function to optimize the distance between the latent variables from the RGB encoder and RAW encoder. With the aforementioned strategies, it is easier to train an encoder for raw images.

3.3. Compressed domain ISP downstream tasks

The compressed domain for raw images is designed and trained with a guided RGB image encoder-decoder, and the ISP downstream tasks are performed in the compressed domain. As a proof-of-concept we discuss the designs for three of these tasks: demosaicing, multi/single -frame (MF/SF) denoising, and registration.

Compressed domain demosaicing. Demosaicing is a critical task in a camera ISP pipeline, and our RGB-guided raw image compression naturally achieves demosaicing in the compressed domain, because the raw image encoder maps the input to the same latent point as the corresponding RGB image encoder. The desired RGB image can be reconstructed by passing the compressed latent variables of the raw image $E_{\mathcal{RAW}}(x_{raw})$ to the RGB decoder $G_{\mathcal{RGB}}$. The loss function in this case can be extended from Eq. (3) to the formula as below,

$$\begin{aligned}\mathcal{L}_{\mathcal{RAW}} = \mathbb{E}_{x_{rgb}\sim p_x}[&\mathcal{L}_1(E_{\mathcal{RAW}}(x_{raw}), E_{\mathcal{RGB}}(x_{rgb})) \\ &+ \lambda_1 \cdot \mathcal{L}_2(x_{rgb}, G_{\mathcal{RGB}}(\mathcal{Q}(E_{\mathcal{RAW}}(x_{raw})))) \\ &+ \lambda_2 \cdot \mathcal{R}(x_{rgb}, G_{\mathcal{RGB}}(\mathcal{Q}(E_{\mathcal{RAW}}(x_{raw})))],\end{aligned} \tag{4}$$

where $\mathcal{R}$ is the perceptual loss from VGG16.

Compressed domain denoising. The compressed latent manifold is expected to preserve noise when reconstructing images from noisy input images. Unlike current non-compression joint demosaicing and denoising algorithms, we choose to employ an additional module in our compressed domain ISP.

Multi-frame averaging is widely used in multi-frame (MF) camera ISP to reduce noise and preserve details. Similarly, in the compressed domain ISP, denoising can be achieved by averaging the compressed latent variables of multiple frames. When multi-frame is not available, we can perform single-frame (SF) AI denoising with a simple Unet in the compressed domain. The loss function is as follows,

$$\mathcal{L}_{denoise} = \mathbb{E}_{x\sim p_x}[\mathcal{L}_1(D(\hat{y_n};\sigma), \hat{y})], \tag{5}$$

where $D(\cdot)$ is the Unet that takes the noisy compressed latent variables as input and generates denoised latent variables. Denote the compressed latent variables as $\hat{y} = \mathcal{Q}(E_{\mathcal{RAW}}(x_{raw}))$ from the clean raw image and $\hat{y_n} = \mathcal{Q}(E_{\mathcal{RAW}}(x_{raw} + \sigma))$ from the noisy raw image. The denoising module is trained for a fixed noise level σ.

Compressed domain affine registration. When the depth of the camera scene is large enough, the motion of camera shaking could be simulated with 2D affine registration consisting of 2D rotation and 2D translation given a short

shutter time. To estimate the motion, the latent variables of an original image and a target image with 2D affine transformation are fed into the registration module, and the three affine parameters are predicted by the registration network. MSE loss is used for all three parameters.

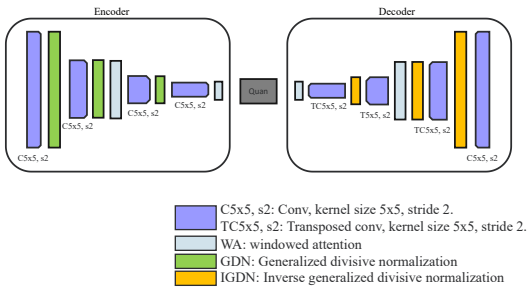

Figure 3. Network architecture of RGB encoder and decoder. Note that raw encoder is the same as the RGB encoder: raw image is zero filled at un-sampled-locations before feeding to the encoder.

4. Experiments

4.1. Experiments setup and implementation details

Training. We implement our proposed compressed domain ISP and RGB-guided training strategy in PyTorch [30]. As shown in Figure 3, we adopt the same RGB encoder and decoder structure with a window-based attention module with GDN [3] as proposed in [54]. Note that we don't include the auto-regressive module for side information.

For RGB image compression, we randomly choose 300k images from the OpenImages dataset [22] and randomly crop them to the size of 256×256 to save GPU memory. We use the Adam optimizer [18] on MSE loss with a batch size of 64. The initial learning rate is set to 1×10^{-4} for 30 epochs, and drops to 3×10^{-5} for the next 30 epochs. There are two experiments with two different channel numbers of latent output: 200 and 120.

For raw image demosaicing, we adopt the same architecture from the RGB encoder and initialize it with the parameters of the RGB encoder. The raw images are converted to RGB by zero-filling pixels before inputting to the network. We apply additional L1 loss between the output of the RGB and the raw encoder to enforce identical mapping between two encoders. We used the same training images from the OpenImages dataset with simulated Bayer and Tetra CFA patterns. Note that for high-resolution images, the Tetra pattern (also known as quad Bayer) is preferred and more widely used compared with the Bayer pattern. We also acquire 2500 high-resolution (25MP) images and crop them into 4 equal size patches as an internal dataset. We randomly select 90% of the patches and add them to the training set. The remaining 10% is used for evaluation. Under the guidance of the RGB encoder, only 5 epochs are needed with an initial learning rate of 1×10^{-4}. We use $\lambda_1 = 0.1$ and $\lambda_2 = 0.1$.

For the compressed domain denoising with a compression ratio of 4, we conduct two scenarios: 1. Multi-frame (MF) with direct blending in the compressed domain. 2. Single-frame (SF) with blind DL denoising module. Signal-dependent Poisson noise with a mean of 1000 is added to the raw image. The architecture is illustrated in Figure S1.

For the registration between two frames, we simulate the 2d affine motion of 2d translation (4% of the height and width) and rotation (-10 degrees to 10 degrees). We train it with the internal high-resolution dataset only. The architecture is shown in figure S2. For both denoising and registration, we train them for 100 epochs with learning rate of 1×10^{-4}.

Comparisons with SOTA/baselines. To the best of our knowledge, we are among the first to propose a phone camera ISP pipeline directly in the compressed domain to address memory limitations with high-resolution images. As a result, there is no direct comparison between our method and existing state-of-the-art (SOTA) methods. However, for RGB reconstruction, we choose WAM [54] as the baseline. For raw image demosaicing, we compare against Deepjoint [12], which operates on noise-free raw images using Bayer pattern without compression.

Evaluation. We evaluate our proposed method for RGB image compression and raw image demosaicing by calculating the average reconstruction performance (PSNR and SSIM) on commonly used public datasets: Kodak image set [20], CLIC validation dataset [1], MCM [51], and our internal DSLR high-resolution dataset. We conduct the experiments under compression ratios of 4 and 6, using both the Tetra pattern and Bayer pattern. For denoising and registration, we evaluate them with our internal dataset only as a proof-of-concept. The compressed domain is motivated and designed to address the issues of high-resolution images.

4.2. Ablation study

We propose an RGB-guided design strategy for raw image compression, incorporating both initialization from the RGB encoder and constraints of mapping to the RGB compressed latent manifold. To better reveal the effectiveness of our proposed strategy, we explore different design strategies for the Tetra pattern, which is widely adopted for high-resolution images:

1. Use RGB guidance only with initialization. This involves initializing the encoder and decoder with parameters from the RGB encoder and decoder, respectively. Instead of mapping the compressed latent manifold to the same manifold defined by RGB compression, we optimize both the raw encoder and RGB decoder specifically for demosaicing.

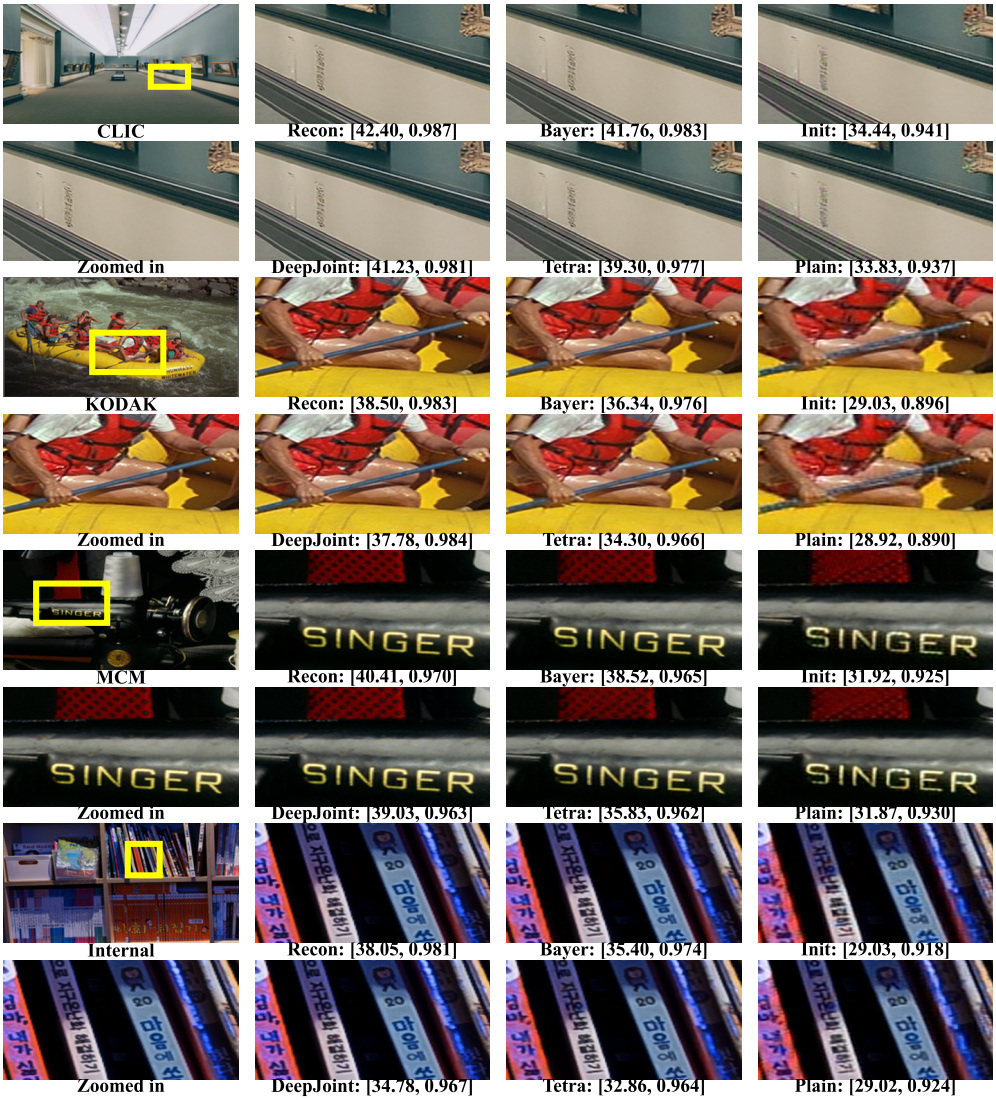

Figure 4. Demosaicing results with a compression ratio of 4 across four datasets (CLIC, KODAK, MCM, and Internal). The baseline comparison is performed using DeepJoint trained on Bayer pattern. Evaluation metrics include [PSNR,SSIM]. A zoomed-in image, focused on the content within the yellow box, is provided for detailed analysis.

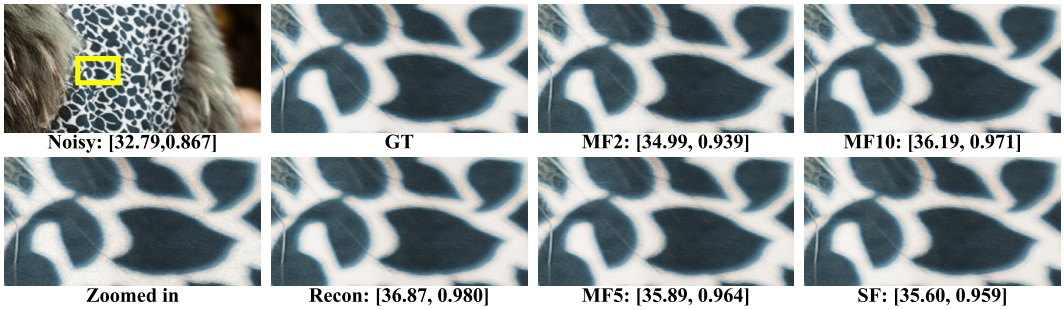

Figure 5. Denoising results on compressed latent manifold. The evaluation metrics are shown as [PSNR, SSIM]. The single frame denoising with learned CNN denoiser achieves similar performance compared with an average of 5 independent frames in the compressed domain.

2. Abandon RGB guidance. Optimize raw encoder and RGB decoder from scratch.

By sharing the same compressed latent manifold, the ISP pipeline could compress RGB images and raw images in a uniform format.

4.3. Visual Quality

Figure 2 shows the RGB compression with (baseline) and without (ours) entropy encoding and auto-regressive module. Our approach outperforms the baseline, showcasing superior reconstruction quality for compressed domain ISP. Figures 4 and S3 display visual examples of demosaicing on datasets with compression ratios of 4 and 6. Our proposed compressed domain demosaicing framework achieves comparable performance to DeepJoint on the majority of images. Notably, while DeepJoint preserves the details of red foams better, our framework may exhibit slight blurring in this aspect, particularly evident in the MCM dataset. Nevertheless, our method generally achieves satisfactory demosaicing performance within the compressed domain. Figure 5 illustrates the denoising ISP performance within the compressed domain. In the multi-frame scenario, conventional blending exhibits improved performance with an increased number of frames. Conversely, in the single-frame scenario, a trained denoising module can effectively eliminate noise directly from the compressed domain, while preserving high-frequency details. Figure 6 depicts the visual outcomes of our high-accuracy registration and warping, as facilitated by our registration module within the compressed domain. In real-world scenarios involving multi-frame motion, the motion range is typically smaller, further highlighting the effectiveness of our approach.

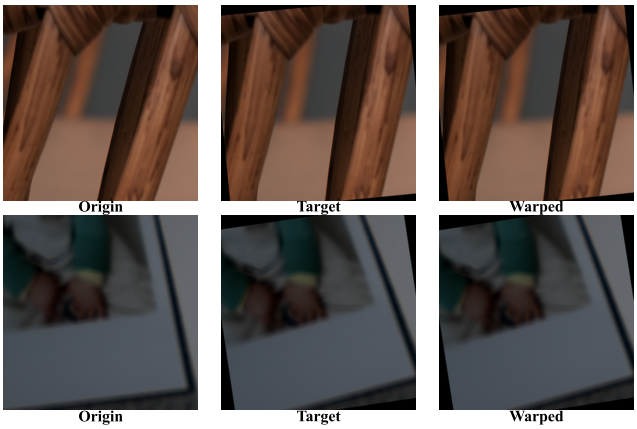

Figure 6. Examples of 2d affine registration results. The registration module on the latent manifold achieves successful results to warp the image from the original image to the target image.

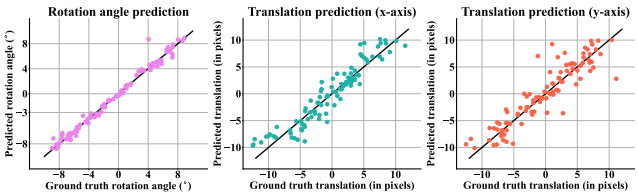

Figure 7. Results of predicted 2d affine parameters from compressed domain and the ground truth.

4.4. Quantitative analysis

Table 1 shows the quantitative results, PSNR and SSIM, for RGB compression and Raw image demosaicing with Bayer and Tetra patterns, under compression ratios at 4 and 6. Without entropy encoding loss, our RGB compression achieves higher metrics. Based on the manifold defined by RGB compression, our compressed domain demosaicing framework achieves a similar performance as the baseline. From the comparisons with ablation studies, our RGB-guided strategy significantly improves the performance.

Table 2 shows the quantitative results of denoising tasks on the compressed domain. Direct multi-frame blending (averaging) on compressed variables can reduce the noise with increased PSNR and SSIM. For a single frame, the denoising module achieves denoising performance roughly equivalent to 4 frames blending.

Figure 7 shows the predicted 2d affine parameters and ground truth. Our registration module can predict high-accuracy registration parameters directly based on compressed variables. For the registration results, in our internal test dataset, the average absolute rotation error is 0.43 degrees and x translation error is 2.8 pixels and the y translation is 3.3 pixels.

4.5. Discussion

Designing the compressed manifold: t is worth noting that we construct the compressed domain without any constraints related to downstream tasks. Our experimental findings in demosaicing, single and multi-frame denoising, and registration reveal that the knowledge acquired within the compressed domain is sufficiently robust to yield satisfactory performance.

Entropy loss consideration: The size of entropy encoding presents a trade-off between the effectiveness of the manifold. Our focus is on designing an effective compressed latent manifold to compress the raw image, thus conserving memory by reducing the input size of the ISP modules. As depicted in Figure 2, optimizing the rate-distortion trade-off using entropy encoding can result in the loss of fine details, which are essential for compressed domain ISP. Furthermore, our experiments reveal that simple quantization-based AI compression is three times faster than entropy en-

Task	Methods	CLIC	KODAK	MCM	Internal
Recon	WAM(x4)	[34.70, 0.934]	[34.90, 0.938]	[34.27, 0.924]	[39.15, 0.963]
	Ours(x4)	[**38.42, 0.974**]	[**39.68, 0.982**]	[**37.42, 0.959**]	[**42.28, 0.984**]
	WAM(x6)	[32.88, 0.890]	[32.54, 0.878]	[32.13, 0.868]	[35.90, 0.913]
	Ours(x6)	[**37.02, 0.961**]	[**37.15, 0.964**]	[**36.73, 0.953**]	[**41.82, 0.981**]
Demosaic(B)	DeepJoint	[**37.30, 0.965**]	[**38.98, 0.980**]	[**35.75**, 0.942]	[40.87, 0.976]
	Ours(x4)	[37.07, **0.965**]	[38.25, 0.976]	[35.71, **0.944**]	[**41.20, 0.981**]
	Ours(x6)	[34.64, 0.943]	[34.65, 0.946]	[34.32, 0.931]	[39.33, 0.971]
Demosaic(T)	Plain(x4)	[31.44, 0.910]	[31.18, 0.912]	[30.21, 0.875]	[36.35, 0.955]
	Init(x4)	[31.70, 0.913]	[31.00, 0.915]	[30.40, 0.883]	[36.47, 0.955]
	Ours(x4)	[**35.50, 0.959**]	[**36.46, 0.969**]	[**33.06, 0.932**]	[**39.35, 0.978**]
	Plain(x6)	[28.98, 0.812]	[27.89, 0.823]	[28.35, 0.826]	[33.87, 0.916]
	Init(x6)	[31.26, 0.906]	[30.80, 0.906]	[30.10, 0.873]	[36.32, 0.953]
	Ours(x6)	[**33.65, 0.936**]	[**33.83, 0.939**]	[**32.38, 0.914**]	[**38.12, 0.966**]

Table 1. Quantitative metrics on RGB compression reconstruction, Bayer (B) raw image demosaicing and Tetra (T) raw image demosaicing. The numbers in the table are the fashion of [PSNR, SSIM]. Note that our proposed method achieves the best performance compared with the ablation experiments and similar performance with DeepJoint (trained in image domain) in demosaicing task.

coding with auto-regressive modules. Therefore, we omit the entropy encoding rate cost and use reconstruction loss only to improve the capability of the compressed latent manifold.

Analysis of design degrees of freedom: Our ablation studies explore the functional guidance provided by RGB image compression. Our proposed method fully utilizes the assistance of RGB compression. Compared to design choices without RGB guidance, whether using initialization or not, our methods achieve superior results. This suggests that the constraints imposed by RGB guidance are more influential than the design choices themselves. We can interpret this as a form of regularization, making optimization easier to converge into better solutions.

Analysis of denoising ISP task: In our experiments, we simulate camera noise using Poisson noise, although a potentially better choice could involve mixed Gaussian-Poisson noise. In this work, we treat denoising as a supervised blind denoising task, training the module for single-frame denoising or employing simple blending for multi-frame denoising. However, single-image denoising is limited by the reduced spatial information due to downsampling operators in the encoder, while blending is constrained by the mismatch between the assumed i.i.d. zero-mean distribution and the true distribution in the compressed manifold. Modeling the noise distribution, especially for Poisson noise, remains a topic for future study, which could further enhance the performance of the denoising module.

Registration choice: In this work, we illustrate the potential of our compressed domain ISP for downstream tasks by designing a module for two-frame 2D affine registration. It's worth noting that our current design does not account for rotation and translation invariant. Consequently, warping needs to be performed in the image domain rather than the compressed domain. Future work could focus on designing a method to achieve compressed domain warping by incorporating constraints during the construction of the compressed domain manifold.

Metrics	Clean	Noisy	MF2	MF5	MF10	SF
PSNR	39.35	35.68	37.17	38.21	38.58	37.69
SSIM	0.978	0.918	0.949	0.965	0.970	0.961

Table 2. Quantitative metrics on compressed domain denoising task with Poisson noise level at K = 1000 in the raw image domain. MF represents multi-frame averaging. SF represents single-frame denoising.

5. Conclusion

In this work, we introduce a novel concept aimed at realizing a mobile camera ISP pipeline directly on the compressed domain to tackle the challenges posed by high memory consumption from super-high-resolution images. Our approach involves a unique design strategy where we utilize the same compressed manifold designed in RGB image compression and train the raw image encoder with guidance from the RGB encoder. We show the capability of this compressed domain ISP by implementing demosaicing, single-frame denoising, multi-frame denoising, and registration downstream tasks. Both quantitative and visual results provide compelling evidence that our proposed compressed domain ISP is feasible and paves the way for further exploration in future studies.

References

[1] Workshop and challenge on learned image compression. https://www.compression.cc, 2020. 5

[2] Yamin Arefeen, Junshen Xu, Molin Zhang, Zijing Dong, Fuyixue Wang, Jacob White, Berkin Bilgic, and Elfar Adalsteinsson. Latent signal models: Learning compact representations of signal evolution for improved time-resolved, multi-contrast mri. *Magnetic Resonance in Medicine*, 2023. 1

[3] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. Density modeling of images using a generalized normalization transformation. *arXiv preprint arXiv:1511.06281*, 2015. 5

[4] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016. 1, 2

[5] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018. 1, 2

[6] Fabrice Bellard. Bpg image format (2014). *URL http://bellard. org/bpg/.[Online, Accessed 2016-08-05]*, 1 (2), 2016. 1

[7] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7939–7948, 2020. 1, 2

[8] Yoojin Choi, Mostafa El-Khamy, and Jungwon Lee. Variable rate deep image compression with a conditional autoencoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3146–3154, 2019. 1

[9] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015. 1

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[11] Pascal Getreuer, Ignacio Garcia-Dorado, John Isidoro, Sungjoon Choi, Frank Ong, and Peyman Milanfar. Blade: Filter learning for general purpose computational photography. In *2018 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11. IEEE, 2018. 2

[12] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 2, 5

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[14] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14771–14780, 2021. 1

[15] Xiang Huang and Oliver Cossairt. Dictionary learning based color demosaicing for plenoptic cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 449–454, 2014. 2

[16] Andrey Ignatov, Luc Van Gool, and Radu Timofte. Replacing mobile camera isp with a single deep learning model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 536–537, 2020. 1

[17] Wooseok Jeong and Seung-Won Jung. Rawtobit: A fully end-to-end camera isp network. In *European Conference on Computer Vision*, pages 497–513. Springer, 2022. 4

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1

[20] Eastman Kodak. Kodak lossless true color image suite (photocd pcd0992). http://r0k.us/graphics/kodak/, 1993. 5

[21] Filippos Kokkinos and Stamatios Lefkimmiatis. Deep image demosaicking using a cascade of convolutional residual denoising networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 303–319, 2018. 2

[22] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. 5

[23] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. *arXiv preprint arXiv:1809.10452*, 2018. 2

[24] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*, 2018. 1

[25] Zhetong Liang, Jianrui Cai, Zisheng Cao, and Lei Zhang. Cameranet: A two-stage framework for effective camera isp learning. *IEEE Transactions on Image Processing*, 30:2248–2262, 2021. 1

[26] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4394–4402, 2018. 2

[27] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, 33:11913–11924, 2020. 2

[28] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2502–2510, 2018. 1

[29] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018. 2

[30] 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. 5

[31] Ibrahim Pekkucuksen and Yucel Altunbasak. Gradient based threshold free color filter array interpolation. In *2010 IEEE International Conference on Image Processing*, pages 137–140. IEEE, 2010. 2

[32] Aakanksha Rana, Praveer Singh, Giuseppe Valenzise, Frederic Dufaux, Nikos Komodakis, and Aljosa Smolic. Deep tone mapping operator for high dynamic range images. *IEEE Transactions on Image Processing*, 29:1285–1298, 2019. 1

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[34] Eli Schwartz, Raja Giryes, and Alex M Bronstein. Deepisp: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing*, 28(2):912–923, 2018. 1

[35] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001. 1

[36] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018. 1

[37] Nai-Sheng Syu, Yu-Sheng Chen, and Yung-Yu Chuang. Learning deep convolutional networks for demosaicing. *arXiv preprint arXiv:1802.03769*, 2018. 1

[38] Hanlin Tan, Xiangrong Zeng, Shiming Lai, Yu Liu, and Maojun Zhang. Joint demosaicing and denoising of noisy bayer images with admm. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2951–2955. IEEE, 2017. 1

[39] David S Taubman, Michael W Marcellin, and Majid Rabbani. Jpeg2000: Image compression fundamentals, standards and practice. *Journal of Electronic Imaging*, 11(2): 286–287, 2002. 2

[40] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017. 2

[41] George Toderici, Sean M O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. *arXiv preprint arXiv:1511.06085*, 2015. 2

[42] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5306–5314, 2017. 2

[43] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014. 1

[44] Michael Tschannen, Eirikur Agustsson, and Mario Lucic. Deep generative models for distribution-preserving lossy compression. *Advances in neural information processing systems*, 31, 2018. 2

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[46] Gregory K Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991. 1

[47] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 1

[48] Wenzhu Xing and Karen Egiazarian. End-to-end learning for joint image demosaicing, denoising and super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3507–3516, 2021. 1

[49] Junshen Xu, Molin Zhang, Esra Abaci Turk, Larry Zhang, P Ellen Grant, Kui Ying, Polina Golland, and Elfar Adalsteinsson. Fetal pose estimation in volumetric mri using a 3d convolution neural network. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part IV 22*, pages 403–410. Springer, 2019. 1

[50] Junshen Xu, Molin Zhang, Esra Abaci Turk, P Ellen Grant, Polina Golland, and Elfar Adalsteinsson. 3d fetal pose estimation with adaptive variance and conditional generative adversarial network. In *Medical Ultrasound, and Preterm, Perinatal and Paediatric Image Analysis: First International Workshop, ASMUS 2020, and 5th International Workshop, PIPPI 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4-8, 2020, Proceedings 1*, pages 201–210. Springer, 2020. 1

[51] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic imaging*, 20(2): 023016–023016, 2011. 5

[52] Molin Zhang, Junshen Xu, Esra Abaci Turk, P Ellen Grant, Polina Golland, and Elfar Adalsteinsson. Enhanced detection of fetal pose in 3d mri by deep reinforcement learning with physical structure priors on anatomy. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VI 23*, pages 396–405. Springer, 2020. 1

[53] Molin Zhang, Junshen Xu, Yamin Arefeen, and Elfar Adalsteinsson. Zero-shot self-supervised joint temporal image and sensitivity map reconstruction via linear latent space. In *Medical Imaging with Deep Learning*, pages 1713–1725. PMLR, 2024. 1

[54] Renjie Zou, Chunfeng Song, and Zhaoxiang Zhang. The devil is in the details: Window-based attention for image

compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17492–17501, 2022. 1, 5