CRA: A GENERIC COMPRESSION RATIO ADAPTER FOR END-TO-END DATA-DRIVEN IMAGE COMPRESSIVE SENSING RECONSTRUCTION FRAMEWORKS

Zhikang Zhang, Kai Xu, Fengbo Ren

Parallel Systems and Computing Laboratory
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University

ABSTRACT

End-to-end data-driven image compressive sensing reconstruction (EDCSR) frameworks achieve state-of-the-art reconstruction performance in terms of reconstruction speed and accuracy. However, due to their end-to-end nature, existing EDCSR frameworks can not adapt to a variable compression ratio (CR). For applications that desire a variable CR, existing EDCSR frameworks must be trained from scratch at each CR, which is computationally costly and time-consuming. This paper presents a generic compression ratio adapter (CRA) framework that addresses the variable compression ratio (CR) problem for existing EDCSR frameworks with no modification to given reconstruction models nor enormous rounds of training needed. CRA exploits an initial reconstruction network to generate an initial estimate of reconstruction results based on a small portion of the acquired measurements. Subsequently, CRA approximates full measurements for the main reconstruction network by complementing the sensed measurements with resensed initial estimate. Our experiments based on two public image datasets (CIFAR10 and Set5) show that CRA provides an average of 13.02 dB and 5.38 dB PSNR improvement across the CRs from 5 to 30 over a naive zero-padding approach and the AdaptiveNN approach(a prior work), respectively. CRA addresses the fixed-CR limitation of existing EDCSR frameworks and makes them suitable for resource-constrained compressive sensing applications.

Index Terms— compressive sensing, neural network, signal processing, estimate resensing

1. INTRODUCTION

Compressive sensing (CS) is a signal sensing technique that senses signals in a compressed manner to save sensing and transmission costs [1, 2]. The sensing in CS is a simple linear mapping of the original signal, and the reconstruction in CS is a complicated inverse problem. Most existing CS reconstruction methods [3, 4, 5, 6, 7, 8] formulate the reconstruction process as an optimization problem and search for the solution iteratively. We refer to them as iterative reconstruction methods. Recently, as neural networks have been proven to be powerful tools for approximation and generation tasks, many neural network models [9, 10, 11] have been proposed to approximate the inverse mapping of CS directly. We refer to the neural network models that directly map CS measurements to reconstruction results as end-to-end data-driven CS reconstruction (EDCSR) frameworks. Compared with the conventional iterative reconstruction methods, the EDCSR frameworks offer significant improvements on both reconstruction speed and accuracy, especially at high compression ratios(CRs) [9, 11], establishing the possibility to perform real-time, high-accuracy image CS reconstruction [12].

Allowing for a variable CR that can be adaptive to the available battery level, storage space, or communication bandwidth at run time is critical to many resource-constrained CS applications [13, 14, 15]. Unfortunately, a major limitation of the existing EDCSR frameworks is that they can only perform reconstruction at fixed CRs once they are trained. For reconstruction at a different CR, an EDCSR framework must be trained at that CR from scratch, which greatly limits their application in variable CR scenarios.

In this paper, we propose to apply the concept of estimate resensing to empower EDCSR frameworks with the adaptability to variable CR. Our approach is structured as a generic CR adapter (CRA) that can be independently applied to the existing EDCSR frameworks with no modification to a given reconstruction model nor enormous rounds of training needed. Given a user-defined lower and upper bounds of the CR, CRA exploits an initial reconstruction network which is trained at the highest CR to generate an initial estimate of reconstruction results with the sensed measurements. Subsequently, CRA approximates full measurements for the main reconstruction network, which is trained at the lowest CR, by complementing the sensed measurements available at any intermediate CR with resensed initial estimate. As such, CRA

This work is supported by a NSF grant (IIS/CPS-1652038) and an unrestricted gift (CG#1319167) from Cisco Research Center. The NVIDIA TITAN X GPUs used for this research were donated by the NVIDIA Corporation.

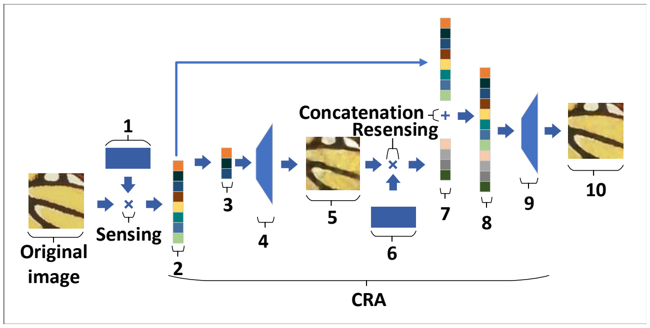

Fig. 1. **The data flow of the sensing and reconstruction process of CRA. 1**: first m rows of the sensing matrix A. **2**: sensed measurements $Y_{1\to m}$. **3**: first m_{min} measurements. **4**: initial reconstruction network f_{init}. **5**: initial estimate X'. **6**: last $m_{max} - m$ rows of the sensing matrix A. **7**: pseudo measurements $Y'_{m+1\to m_{max}}$. **8**: concatenated measurement vector $\hat{Y}$. **9**: main reconstruction network f_{main}. **10**: reconstruction result $\hat{X}$. Details are in Section 3.

can enable flexible reconstruction with an arbitrary number of measurements and extend the supported CR to a user-defined lower and upper bounds at a fine granularity. The main advantage of CRA is that it is generic and provides an approximately linear trade-off between the number of measurements and the reconstruction accuracy for all EDCSR frameworks.

The contributions of this paper are two-fold. First, we propose a simple yet effective approach to empower EDCSR frameworks with adaptability to variable CR, which makes them suitable for resource-constrained CS application scenarios. The proposed CRA significantly improves the reconstruction accuracy of the existing EDCSR frameworks in the context of variable CR compared to a naive zero-padding approach and the prior work [16]. Second, our approach is generic for all EDCSR frameworks and can empower them to deal with a variable CR at run time with no modification to the given network model nor enormous training time needed.

2. RELATED WORK

2.1. Iterative Reconstruction Methods

Most of the existing reconstruction methods of CS are iterative reconstruction methods [3, 4, 5, 6, 7, 8]. Iterative reconstruction methods can inherently adapt to a variable CR but are limited by their low reconstruction speed due to their iterative nature as well as low reconstruction accuracy at high CRs. [11] shows empirically that most of the iterative reconstruction methods have lower reconstruction accuracy at high CRs compared with EDCSR methods.

2.2. Rate-adaptive Neural Network(AdaptiveNN)

To the best of our knowledge, AdaptiveNN [16] is the only work so far that aims to solve the variable CR problem for EDCSR frameworks. AdaptiveNN proposes to constrain the first layer of an EDCSR framework to be the pseudo inverse of the sensing matrix during the training. The main limitations of AdaptiveNN are low reconstruction accuracy, long training time needed, and the lack of generality. Overcoming these limitations, the proposed CRA approach achieves more than 20% higher reconstruction accuracy (Fig 2, Fig 3) with 75x less training training time (Table 1) compared to AdaptiveNN. Moreover, CRA is generic and can be applied to all EDCSR frameworks.

3. METHODOLOGY

Gaussian random sensing matrices are used in this work as they are the most widely used sensing matrices in CS related studies. Assume the original signal is a n-dimensional vector $X = [x_1, \cdots, x_n]$. The user-defined lower and upper bounds of CR are $CR_{min} = \frac{n}{m_{max}}, CR_{max} = \frac{n}{m_{min}}$. Conventionally, for a signal that has to be sensed at the CR $\frac{n}{m}, m_{min} \leq m \leq m_{max}$, the sensing step is a linear transformation of the signal, i.e. $Y = AX$, where A denotes a sensing matrix in the size of m by n, and $Y = [y_1, \cdots, y_m]$ denotes the compressively sensed measurements of X. The corresponding EDCSR network that is trained at the CR $\frac{n}{m}$ with A is essentially a high-dimensional, vector-valued function that maps a m-dimensional space to a n-dimensional space, i.e. $\hat{X} = f(Y, \Theta)$, where f is the reconstruction franework with trainable parameters Θ, and $\hat{X} = [\hat{x}_1, \cdots, \hat{x}_n]$ is the reconstruction result. The sensing matrix A is predefined before the training of f and the trainable parameters Θ are fixed once the network is trained.

The overall process of sensing and reconstruction with CRA is shown in Fig 1. A random sensing matrix A in the size of m_{max} by n is predefined. Two EDCSR frameworks named initial reconstruction network(f_{init}) and main reconstruction network(f_{main}) are pretrained at CR_{max} and CR_{min} with the first m_{min} rows of A and all rows of A, respectively. For performing the sensing and reconstruction of a signal X at arbitrary $CR = \frac{n}{m}$ between CR_{min} and CR_{max}, the first m row of A are used to sense X to get measurements $Y = [y_1, \cdots, y_m]$. CRA adopts f_{init} to generate an initial estimate $X' = [x'_1, \cdots, x'_n]$ of the signal by taking the first m_{min} sensed measurements $Y_{1\to m_{min}} = [y_1, \cdots, y_{m_{min}}]$ as input. Subsequently, $m_{max} - m$ additional pseudo measurements $Y'_{m+1\to m_{max}} = [y'_{m+1}, \cdots, y'_{m_{max}}]$ of the signal X are generated by resensing the initial estimate X' with the last $m_{max} - m$ rows of A. Finally, the full measurements at the CR_{min} are approximated by concatenating sensed measurements $Y_{1\to m}$ and pseudo measurments $Y'_{m+1\to m_{max}}$ to $\hat{Y} = [y_1, \cdots, y_m, y'_{m+1}, \cdots, y'_{m_{max}}]$. As such, regardless of

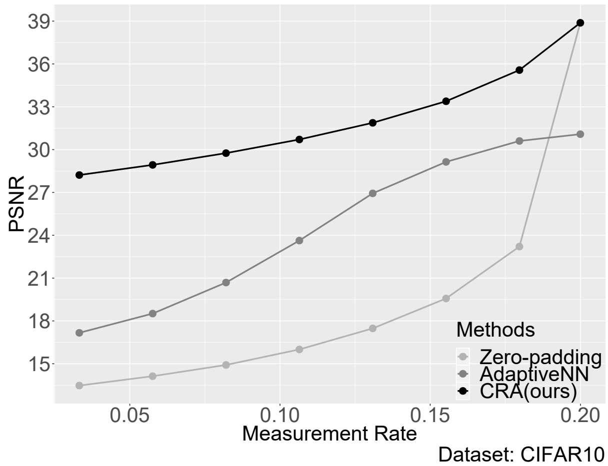

Fig. 2. **Reconstruction accuracy comparison on CIFAR10**

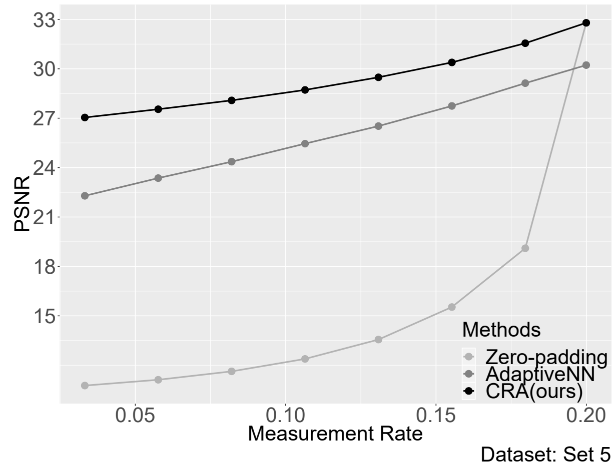

Fig. 3. **Reconstruction accuracy comparison on Set 5**

the CR at run-time, CRA can always provide approximated full measurements $\hat{Y}$ that can be directly fed into f_{main} for the final reconstruction of the signal, i.e. $\hat{X} = f_{main}(\hat{Y}, \Theta)$.

4. EXPERIMENTS

The common setups of all the experiments are shown below. The CR_{max} and CR_{min} are set to 30 and 5, respectively. We conduct two sets of experiments on different datasets. The first set of experiments uses CIFAR10 [17](resized to 64 by 64) for both training and testing. The second set of experiments uses the dataset made by [11] for training and Set5(cut into 64 by 64 blocks) [18] for testing. For each sample, compressive sensing with the same sensing matrix is performed for each RGB channel(The 2-D tensor of each channel is row-wise vectorized to a 4096-dimensional vector before sensing). The reconstruction is performed using the measurements of all three channels. The neural network library used is Pytorch [19]. The EDCSR frameworks used in the experiments are ReconNet[9] and LAPRAN[11]. For the simplicity of illustration, the experiment results are plotted with respect to measurement rate (MR), which is defined as $MR = \frac{1}{CR}$. For each training dataset, 5% of the training samples are randomly selected as the validation set to avoid over-fitting. The model is tested on the validation set at the end of each training iteration. The model offers the best performance on the validation set is used for the final testing.

4.1. Comparison With Existing Solutions

To demonstrate the effectiveness of CRA, we compare the reconstruction accuracy against a modified version of AdaptiveNN[16] (the only prior work to the best of our knowledge), and a naive zero-padding approach. As the source codes of AdaptiveNN are unavailable, we re-implement the AdaptiveNN model based on its original paper using

Approach	MultiNN	AdaptiveNN	CRA
per epoch (second)	22	73	22
num of epoches	204900	7430	600
in total (hour)	1252	150	4

Table 1. **Training time comparison on CIFAR10.** For fair comparison, ReconNet is used in all approaches. Each experiment is done on a single Nvidia GTX Titan X GPU.

PyTorch. We find that the performance of AdaptiveNN is unexpectedly low and can be improved with some simple modifications. Specifically, we add the batch normalization layer [20] right behind each convolution layer and switch the activation function from ReLU to Tanh. Experiments on the CIFAR10 dataset show that the modified AdaptiveNN achieves more than 10000% and 10% improvement on the training speed and reconstruction accuracy, respectively, over the original AdaptiveNN. Zero-padding is a naive approach. Given the EDCSR framework that is trained at the lowest CR, the measurements that are sensed at a higher CR are simply complemented by zeros. The experiments results (Fig 2 and 3) on the CIFAR10 and Set5 datasets show that CRA can achieve a 2.56-11.05dB and 0-16.45dB peak signal-to-noise ratio (PSNR) improvement over the AdaptiveNN and zero-padding approach, respectively, across the MRs of 0.33-0.2(CRs of 30-5).

In theory, the accuracy-optimal approach for EDCSR frameworks to handle variable CR at run time is to train multiple reconstruction networks at each CR needed separately. We refer to this brute-fource approach as MultiNN. In practice, the MultiNN approach is often impractical or unaffordable due to the enormous training time and computational resources required. Table 1 compares the total training time of MultiNN, AdaptiveNN, and CRA needed for handling a variable CR (from 5 to 30) at a fine granularity (step size of

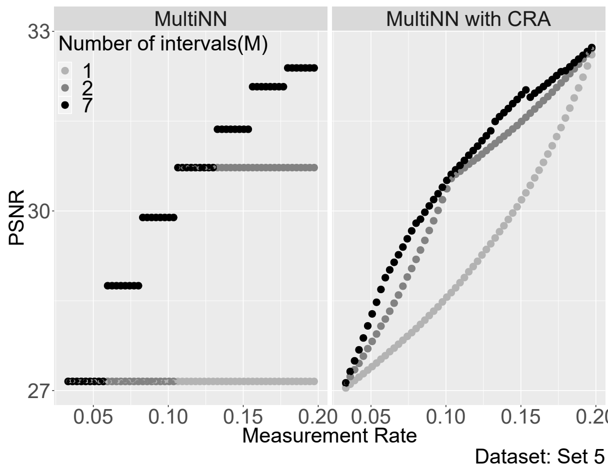

Fig. 4. **Comparing MultiNN with CRA and MultiNN**

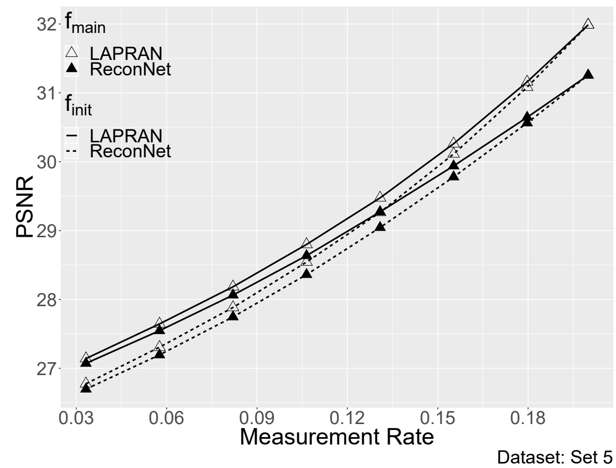

Fig. 5. **Comparing EDCSR framework combinations**

m is 1). The num of epoches of MultiNN is estimated as $\frac{(m_{max}-m_{min})}{\text{step size of } m} \times 300$, where $m_{max} = \frac{n}{CR_{min}} = \frac{4096}{5} = 819$, $m_{min} = \frac{n}{CR_{max}} = \frac{4096}{30} = 136$ and 300 is the predefined number of training iterations of a single ReconNet. The per epoch training time of ReconNet is estimated by averaging the per epoch training time of ReconNets trained at CRs of 5 and 30. The experiment results show that the training time of CRA is 99.5% and 97% less than MultiNN and AdaptiveNN, respectively. Specifically, MultiNN takes more than 52 days to train in this case.

4.2. Combining MultiNN and CRA

Interestingly, one can combine the MultiNN approach with CRA to further improve the reconstruction accuracy with reasonable additional training time. The key is to divide the MR values between the lowest and highest MR into M non-overlapping intervals. For each interval i, a corresponding initial reconstruction network f^i_{init} and a main reconstruction network f^i_{main} are trained at the lowest MR and the highest MR of the interval, respectively. For an arbitrary MR (equivalently CR), one should find the interval j that the MR belongs to and use the corresponding f^j_{init} and f^j_{main} to reconstruct the signal. Additionally, as the highest MR of each interval i is equal to the lowest MR of next interval $i+1$, each f^i_{main}can be used as f^{i+1}_{init}. Consequently, the total number of EDCSR frameworks to be trained is $M + 1$. Total training time is proportional to the number of frameworks to be trained.

To illustrate the impact of combining CRA with MultiNN, we compare the accuracy-MR trade-off curve between MultiNN only and MultiNN combined with CRA for the cases of 1, 2, and 7 intervals in Fig 4. It is shown that the MultiNN approach with a small number of intervals (reasonable training time) can only provide a piece-wise constant approximation of the theoretically optimal accuracy-MR trade-off curve. Differently, MultiNN combined with CRA is able to provide a piece-wise linear approximation of the theoretically optimal accuracy-MR trade-off curve for EDCSR frameworks to handle variable CR at run time.

4.3. Combining Two Different EDCSR Frameworks

Since CRA is generic, one can also adopt two different EDCSR models as the f_{init} and f_{main}, respectively. We are interested in how do different EDCSR models affect the reconstruction performance of CRA. Fig. 5 shows the reconstruction performance of combining ReconNet and LAPRAN[11] by applying the CRA in all possible combinations. It is shown that for low and high MRs, the reconstruction accuracy is more determined by choice of the EDCSR model as the initial and the main reconstruction network, respectively.

5. CONCLUSION

CRA is a simple yet effective approach to address the variable CR problem of the existing EDCSR frameworks. By using an initial reconstruction network to provide an initial estimate of the reconstruction result based on the acquired measurements and generate additional pseudo measurements by resensing the initial estimate is the key to enabling flexible, accurate reconstruction at an arbitrary measurement size. CRA is generic and can leverage the superior reconstruction speed and accuracy of any existing EDCSR framework to deal with variable CR at run time with no modification to the given network models nor long additional training time required. Our experiments on two public datasets show that CRA provides an average of 13.02 dB and 5.38 dB PSNR improvement across the CRs of 5-30 comparing with a naive zero-padding approach and the prior work [16], respectively. The proposed CRA approach addresses a big limitation of the existing EDCSR frameworks and makes them suitable for resource-constrained application scenarios.

6. REFERENCES

[1] Ruitao Xie and Xiaohua Jia, "Transmission-efficient clustering method for wireless sensor networks using compressive sensing," *IEEE transactions on parallel and distributed systems*, vol. 25, no. 3, pp. 806–815, 2013.

[2] Zhen Gao, Linglong Dai, Shuangfeng Han, I Chih-Lin, Zhaocheng Wang, and Lajos Hanzo, "Compressive sensing techniques for next-generation wireless communications," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 144–153, 2018.

[3] P. S. Huggins and S. W. Zucker, "Greedy basis pursuit," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3760–3772, July 2007.

[4] Wotao Yin Weihong Guo, "Edgecs: edge guided compressive sensing reconstruction," *Proc.SPIE*, vol. 7744, pp. 7744 – 7744 – 10, 2010.

[5] Chengbo Li, Wotao Yin, Hong Jiang, and Yin Zhang, "An efficient augmented lagrangian method with applications to total variation minimization," *Computational Optimization and Applications*, vol. 56, no. 3, pp. 507–530, 2013.

[6] Weisheng Dong, Guangming Shi, Xin Li, Yi Ma, and Feng Huang, "Compressive sensing via nonlocal low-rank regularization," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3618–3632, 2014.

[7] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis, "Compressed sensing using generative models," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 537–546.

[8] David Van Veen, Ajil Jalal, Eric Price, Sriram Vishwanath, and Alexandros G Dimakis, "Compressed sensing with deep image prior and learned regularization," *arXiv preprint arXiv:1806.06438*, 2018.

[9] Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga, Ronan Kerviche, and Amit Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 449–458.

[10] Chris Metzler, Ali Mousavi, and Richard Baraniuk, "Learned d-amp: Principled neural network based compressive image recovery," in *Advances in Neural Information Processing Systems*, 2017, pp. 1772–1783.

[11] Kai Xu, Zhikang Zhang, and Fengbo Ren, "Lapran: A scalable laplacian pyramid reconstructive adversarial network for flexible compressive sensing reconstruction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 485–500.

[12] Xin Yuan and Yunchen Pu, "Parallel lensless compressive imaging via deep convolutional neural networks," *Optics express*, vol. 26, no. 2, pp. 1962–1977, 2018.

[13] S. H. Safavi and F. Torkamani-Azar, "Sparsity-aware adaptive block-based compressive sensing," *IET Signal Processing*, vol. 11, no. 1, pp. 36–42, 2017.

[14] M. Leinonen, M. Codreanu, and M. Juntti, "Distributed variable-rate quantized compressed sensing in wireless sensor networks," in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, July 2016, pp. 1–5.

[15] V. Behravan, N. E. Glover, R. Farry, P. Y. Chiang, and M. Shoaib, "Rate-adaptive compressed-sensing and sparsity variance of biomedical signals," in *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, June 2015, pp. 1–6.

[16] Suhas Lohit, Rajhans Singh, Kuldeep Kulkarni, and Pavan Turaga, "Rate-adaptive neural networks for spatial multiplexers," *arXiv preprint arXiv:1809.02850*, 2018.

[17] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., Citeseer, 2009.

[18] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *British Machine Vision Conference (BMVC)*, 2012, pp. 135.1–135.10.

[19] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[20] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.