

# SINCO: A NOVEL STRUCTURAL REGULARIZER FOR IMAGE COMPRESSION USING IMPLICIT NEURAL REPRESENTATIONS

Harry Gao\*, Weijie Gan\*, Zhixin Sun, and Ulugbek S. Kamilov

Computational Imaging Group, Washington University in St. Louis, MO 63130, USA

## ABSTRACT

Implicit neural representations (INR) have been recently proposed as deep learning (DL) based solutions for image compression. An image can be compressed by training an INR model with fewer weights than the number of image pixels to map the coordinates of the image to corresponding pixel values. While traditional training approaches for INRs are based on enforcing pixel-wise image consistency, we propose to further improve image quality by using a new structural regularizer. We present *structural regularization for INR compression (SINCO)* as a novel INR method for image compression. SINCO imposes structural consistency of the compressed images to the groundtruth by using a segmentation network to penalize the discrepancy of segmentation masks predicted from compressed images. We validate SINCO on brain MRI images by showing that it can achieve better performance than some recent INR methods.

**Index Terms**— Image compression, implicit neural representations, neural radiance fields, deep learning.

## 1. INTRODUCTION

Image compression is an important step for enabling efficient transmission and storage of images in many applications. It is widely used in biomedical imaging due to the high-dimensional nature of data. While traditional image compression methods are based on fixed image transforms [1, 2], deep learning (DL) has recently emerged as a powerful data-driven alternative. The majority of DL-based compression methods are based on training autoencoders to be invertible mappings from image pixels to quantized latent representations [3–5].

In this work, we seek an alternative to the autoencoder-based compression methods by focusing on a recent paradigm using *implicit neural representations (INRs)*. INR refers to a class of DL techniques that seek to learn a mapping from input coordinates (e.g.,  $(x, y)$ ) to the corresponding physical quantities (e.g., density at  $(x, y)$ ) by using a *multi-layer perceptron (MLP)* [6–9]. Recent studies have shown the potential of INR in image compression [10–14]. The key idea

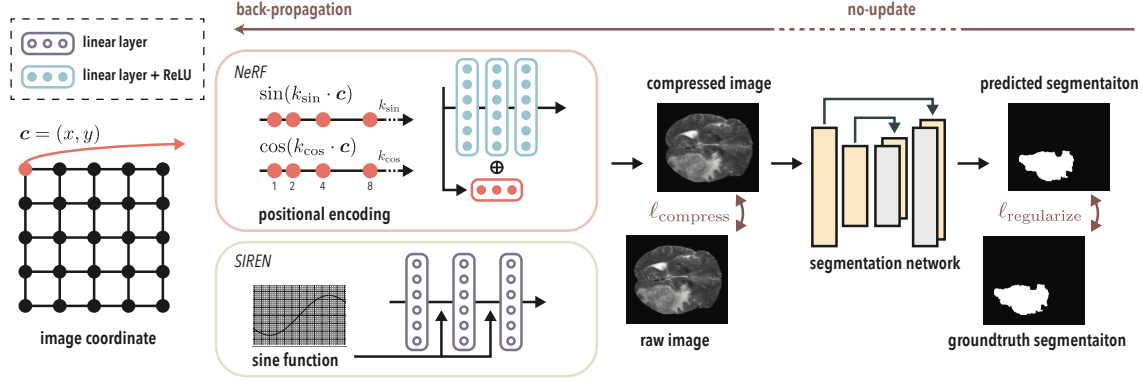
behind INR based compression is to train a MLP to represent an image and consider the weights of the trained model as the compressed data. One can then reconstruct the image by evaluating the pre-trained MLP on the desired pixel locations. The traditional training strategy for image compression using INRs seeks to enforce image consistency between predicted and groundtruth image pixels. On the other hand, it is well-known that image quality can be improved by infusing prior knowledge on the desired images [15, 16]. Based on this observation, we propose *Structural regularization for INR Compression (SINCO)* as new method for improving INR-based image compression using a new structural regularizer. Our structural regularizer seeks to improve the Dice score between the groundtruth segmentation maps and those obtained from the INR compressed image using a pre-trained segmentation network. We validate SINCO on brain MR images by showing that it can lead to significant improvements over the traditional INR-based image compression methods. We show that the combination of the traditional image-consistency loss and our structural regularizer enables SINCO to learn an INR that can better preserve desired image features.

## 2. BACKGROUND

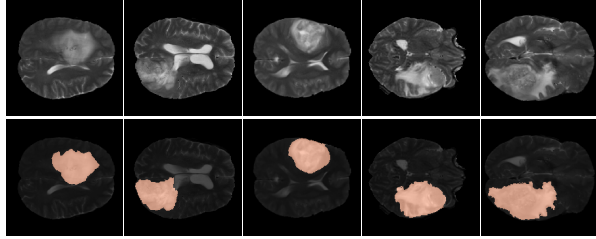
INR (also referred to as *neural fields*) denotes a class of algorithms for continuously representing physical quantities using coordinate-based MLPs (see a recent review [6]). Recent work has shown the potential of INRs in many imaging and vision tasks, including novel view synthesis in 3D rendering [7], video frame interpolation [8], and dynamic imaging [9]. The key idea behind INR is to train a MLP to map spatial coordinates to corresponding *observed* physical quantities. After training, one can evaluate the pre-trained MLP on desired coordinates to predict the corresponding physical quantities, including on locations that were not part of training. Let  $c$  denotes a vector of input coordinates,  $v$  the corresponding physical quantity, and  $\mathcal{M}_\theta$  a MLP with trainable parameter  $\theta \in \mathbb{R}^n$ . The INR training can be formulated as

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^n} \sum_{i=1}^N \ell_{\text{inr}}(\mathcal{M}_\theta(c_i), v_i). \quad (1)$$

\*These authors contributed equally to this work. This material is based upon work supported by the NSF CAREER award under CCF-2043134.



**Fig. 1.** SINCO consists of two components: (a) a multi-layer perceptron (MLP) for compressing an image by mapping its coordinates to the corresponding pixel values; (b) a segmentation network that predicts segmentation masks from the compressed image. Unlike traditional INR methods that only enforce consistency between compressed and groundtruth images, SINCO uses information from a regularizer that penalizes the discrepancy between predicted and groundtruth segmentation masks.



**Fig. 2.** Test images (*top*) and corresponding segmentation masks (*bottom*) used for our numerical evaluations.

where  $N \geq 1$  denotes the number training pairs  $(c, v)$ . The common choices for  $\ell_{\text{inr}}$  include  $\ell_2$  and  $\ell_1$  norms.

INRs have been recently used for image compression [10–13] (see also a recent evaluation in medical imaging [14]). *Compressed Implicit Neural representations (COIN)* [10] is a pioneering work based on training a MLP by mapping the pixel locations (*i.e.*,  $c = (x, y)$ ) of an image to the pixel values. The pre-trained MLP in COIN is quantized and then used as the compressed data. In order to reconstruct the image, one can evaluate the model on the same pixel locations used for training. Several papers have investigated the meta-learning approach to accelerate COIN by first training a MLP over a large collection of datapoints and then fine-tuning it on an instance-dependent one [11, 12, 17]. Two recent papers proposed to regularize INR-based image compression by using  $\ell_0$ - and  $\ell_1$ -norm penalties on the weights of the MLP to improve the compression rate [11, 13].

The structural regularization in SINCO is based on image segmentation using a pre-trained *convolutional neural network (CNN)* (see a comprehensive review of the topic [18]). There exists a rich body of literature in the context of DL-based image segmentation that can be integrated into SINCO [19–21].

To the best of our knowledge, no prior work has considered higher-level structural regularization in the context of INR-based image compression. It is worth mentioning that our structural regularizer is fully compatible to the existing INR compression methods; for example, one can easily combine our structural regularizer with an additional  $\ell_0$ -regularizer.

### 3. PROPOSED METHOD

SINCO consists of two components: (a) a MLP  $\mathcal{M}_\theta$  that represents an image by mapping its coordinates to corresponding pixel values and (b) a CNN  $\mathbf{g}_\varphi$  that predicts a segmentation mask given the compressed image produced by MLP. Specifically, let  $\mathbf{x} \in \mathbb{R}^{H \times W}$  denote an image of height  $H$  and width  $W$  that we seek to compress. Let  $\mathbf{c} \in \mathbb{R}^{HW \times 2}$  represents all the pixel locations within  $\mathbf{x}$ . Then,  $\mathcal{M}_\theta$  is trained to take  $\mathbf{c}$  as input and predict all the corresponding  $HW$  pixels values. We format the output of  $\mathcal{M}_\theta$  to be the compressed image  $\hat{\mathbf{x}} \in \mathbb{R}^{H \times W}$ . The function  $\mathbf{g}_\varphi$  denotes the segmentation CNN that takes the compressed image and predicts a segmentation mask  $\hat{\mathbf{s}} = \mathbf{g}_\varphi(\hat{\mathbf{x}})$ . The SINCO pipeline is illustrated in Fig. 1.

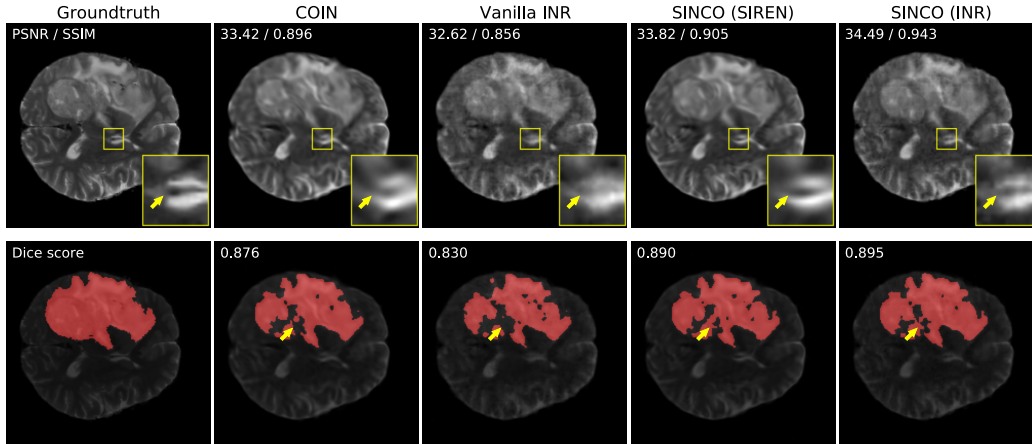
#### 3.1. Network Architecture

We implemented two different architectures for  $\mathcal{M}_\theta$ : (a) SIREN<sup>1</sup> [22] that consists of linear layers followed by sine activation functions and (b) NeRF that incorporates the positional encoding to expand  $\mathbf{c}$  before passing it to  $\mathcal{M}_\theta$ <sup>2</sup> [7]

$$\gamma(\mathbf{c}) = (\sin(2^0 \pi \mathbf{c}), \cos(2^0 \pi \mathbf{c}) \dots \underbrace{\sin(2^{L_f} \pi \mathbf{c})}_{k_{\text{sin}}}, \underbrace{\cos(2^{L_f} \pi \mathbf{c})}_{k_{\text{cos}}}) \quad (2)$$

<sup>1</sup>github.com/lucidrains/siren-pytorch.

<sup>2</sup>github.com/wustl-cig/Coordinate-based-Internal-Learning



**Fig. 3.** Visual illustration of compressed images (*top*) with corresponding segmentation masks (*bottom*) obtained by SINCO and two INR baselines. The PSNR and SSIM values for image compression and dice score values for image segmentation are provided in the top-left corners of the images. This figure shows that SINCO can lead to significant improvements in image quality both quantitatively and qualitatively. Note the quality of the SINCO images in the areas highlighted by the yellow arrow.

where  $L_f > 0$  denotes the number of frequencies.  $\mathcal{M}_\theta$  of NeRF consists of linear layers followed by ReLU activation functions. For the NeRF architecture, we also add residual connections from the input to intermediates layers. We subsequently denote SINCO based on the two MLPs as *SINCO (SIREN)* and *SINCO (INR)*. We adopt the widely-used U-Net [21] architecture as the CNN for the segmentation network  $g_\varphi$ .

### 3.2. Training Strategy

SINCO is trained by minimizing the following loss

$$\ell_{\text{SINCO}} = \ell_{\text{compress}}(\hat{\mathbf{x}}, \mathbf{x}) + \lambda \ell_{\text{regularize}}(\hat{\mathbf{s}}, \mathbf{s}), \quad (3)$$

where  $\mathbf{s}$  denotes the reference segmentation mask of  $\mathbf{x}$ , and  $\lambda \geq 0$  is a parameter that balances image consistency with structural regularization. We implement  $\ell_{\text{compress}}$  as a  $\ell_2$ -norm between the compressed image and the groundtruth image. We implement  $\ell_{\text{regularize}}$  as  $1 - \text{Dice}(\hat{\mathbf{s}}, \mathbf{s})$ , where  $\text{Dice}(\hat{\mathbf{s}}, \mathbf{s})$  is the Dice score coefficient between the segmentation mask predicted from the compressed image and the groundtruth one. In eq. (3), the segmentation network  $g_\varphi$  is assumed to be pre-trained, which implies that we only optimize the parameters of  $\mathcal{M}_\theta$  during training. Note that when  $\lambda = 0$ , eq. (3) is equivalent to the traditional INR-based image compression. After the training, we follow the approach in [10] by quantizing the weights of  $\mathcal{M}_\theta$  from 32-bits to 16-bits.

## 4. NUMERICAL EXPERIMENTS

### 4.1. Setup and comparison

For our experiments, we used MR images of brain tumors with the corresponding segmentation masks obtained from

Metrics	PSNR (dB)	SSIM	Dice score
COIN	34.49	0.907	0.892
Vanilla INR	35.18	0.923	0.877
SINCO (SIREN)	34.93	0.915	0.894
SINCO (INR)	35.69	0.952	0.895

**Table 1.** Average quantitative evaluation metrics over the testing images. This table shows that SINCO can achieve better image compression and segmentation performance than INR baselines by imposing structural regularization.

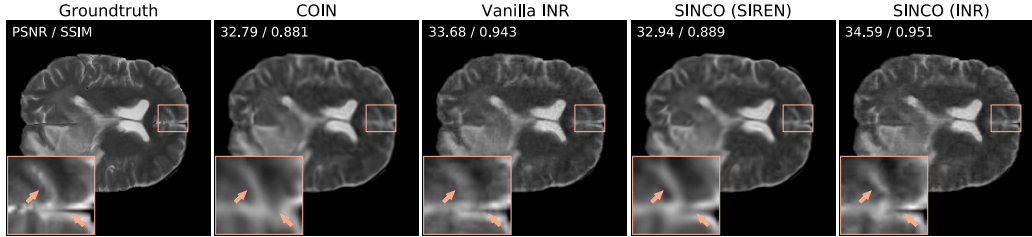
the Decathlon dataset [23] (Task01). We selected ten test images with resolution  $240 \times 240$ . Fig. 2 illustrates several images used in our experiments with the corresponding segmentation masks for tumors.

We compared SINCO against two reference INR methods: (a) *COIN*<sup>3</sup>, a recent INR method discussed in Sec. 2, and (b) *Vanilla INR*, a variant of SINCO (INR) that sets  $\lambda$  in (3) to 0. Note that since COIN also uses SIREN as its MLP architecture, it can be viewed as SINCO (SIREN) using a different loss. For all methods we used the same compression rate quantified with *bits per pixel (bpp)*

$$\text{bpp} = \frac{\text{bit per parameters} \times \#\text{parameters}}{\#\text{pixels}}.$$

We set bpp to 1.2 in our experiments, corresponding to the compression rate of 2% relative to the raw file size. We evaluated all the methods on both image compression and image segmentation using compressed images. For image compression, we used two widely-used quantitative metrics, *peak*

<sup>3</sup>[github.com/EmilienDupont/coin](https://github.com/EmilienDupont/coin)



**Fig. 4.** Visual illustration of compressed images obtained by SINCO and two other INR methods. The PSNR and SSIM values of compressed images relative to the groundtruth are provided in the top-left corner of the images. Note how SINCO provides images with better sharpness and overall quality by leveraging a structural regularizer (see features highlighted by arrows).

signal-to-noise ratio (PSNR), measured in dB, and structural similarity index (SSIM). We used the *Dice score* coefficient to evaluate image segmentation.

#### 4.2. Implementation Details

We followed [21] to train the segmentation network  $g_\varphi$ . We used about 500 images from the same dataset in training. The corresponding training loss can be written as

$$\sum_{i=1}^M \ell_{\text{seg}}(g_\varphi(x_i), s_i) \quad (4)$$

where  $M \geq 1$  denotes the number of training samples and  $\ell_{\text{seg}}$  corresponds to binary cross entropy [21]. We used Adam [24] as an optimizer with learning rate 0.0001. We set training epochs to 75 and batch size to 8. In the training of SINCO, we set  $L_f$  in (2) to 12. We have experimented with different values of  $\lambda$  in (3). The best results were obtained when  $\lambda = 1$ . We used Adam [24] as an optimizer with learning rate 0.001. We set training epochs to 50,000. We performed our experiments on a machine equipped with an Intel Xeon Gold 6130 Processor and an NVIDIA GeForce RTX 1080Ti GPU.

#### 4.3. Results

Figure 3 presents visual results of SINCO and two other INR methods. For image compression, SINCO achieves higher PSNR and SSIM values than other INR methods, while also providing sharper images with features that are more consistent with the groundtruth. For example, SINCO can reconstruct sharper brain tissue highlighted by the yellow arrow, while INR baselines blur out the same features. For image segmentation using compressed images, SINCO can qualitatively and quantitatively outperform INR baselines in general. Especially, SINCO (SIREN) provides more consistent results relative to the groundtruth segmentation mask (see also segmentation mask highlighted by yellow arrows) compared with COIN. Note that COIN also uses SIREN as its MLP architecture. Another set of visual results shown in Figure 4 are also consistent with the observations that SINCO

$\lambda$	0	0.1	0.5	1	2
PSNR (dB)	35.18	35.42	35.45	35.69	35.48
SSIM	0.923	0.948	0.949	0.952	0.950

**Table 2.** Average quantitative evaluation metrics achieved by SINCO (INR) over the test images. The weight of the regularization loss ( $\lambda$ ) is varied, with  $\lambda = 0$  being the Vanilla INR baseline method. This table highlights that the performance of SINCO (INR) can be improved by using the optimized value of  $\lambda$ .

leads to significant improvements in the image quality. Table 1 summarizes quantitative results over all the testing images. The table shows that, under the same MLP architecture (*i.e.*, COIN versus SINCO (SIREN) and Vanilla INR versus SINCO (INR)), SINCO can achieve better performance by leveraging the structural regularization. Table 2 summarizes average quantitative evaluation metrics achieved by SINCO (INR) with different values of  $\lambda$ . Table 2 shows that the performance of SINCO (INR) can be improved by using the optimized value of  $\lambda$ .

## 5. CONCLUSION

We present SINCO as a new structurally regularized image compression method using implicit neural representation. The key idea behind SINCO is to use a pre-trained segmentation network to ensure that the INR compressed images produce accurate segmentation masks. Our experiments on brain MR images show that SINCO can quantitatively and qualitatively outperform traditional INR approaches. In the future work, we will further investigate SINCO for higher dimensional data compression (*e.g.*, 3D or 4D MRI [25]) and leverage recent development of meta-learning strategies to accelerate the INR training [17].

## 6. ACKNOWLEDGEMENTS

The authors are grateful to Yu Sun for helpful discussion.

## 7. REFERENCES

- [1] G. K. Wallace, “The JPEG still picture compression standard,” *Commun. ACM*, vol. 34, no. 4, pp. 30–44, 1991.
- [2] M. Rabbani and R. Joshi, “An overview of the JPEG 2000 still image compression standard,” *Signal Process. Image Commun.*, vol. 17, no. 1, pp. 3–48, 2002.
- [3] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, “Learned Image Compression With Discretized Gaussian Mixture Likelihoods and Attention Modules,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7939–7948.
- [4] D. Minnen, J. Ballé, and G. D. Toderici, “Joint Autoregressive and Hierarchical Priors for Learned Image Compression,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31.
- [5] J. Lee, S. Cho, and S.-K. Beack, “Context-adaptive entropy model for end-to-end optimized image compression,” in *Proc. Int. Conf. Mach. Learn.*, 2018.
- [6] Y. Xie *et al.*, “Neural fields in visual computing and beyond,” *STAR*, vol. 41, no. 2, 2022.
- [7] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “NeRF: Representing scenes as neural radiance fields for view synthesis,” in *European Conf. on Comput. Vis.*, 2020, pp. 405–421.
- [8] W. Shangguan, Y. Sun, W. Gan, and U. S. Kamilov, “Learning Cross-Video Neural Representations for High-Quality Frame Interpolation,” in *European Conf. on Comput. Vis.*, 2022.
- [9] A. W. Reed *et al.*, “Dynamic CT Reconstruction from Limited Views with Implicit Neural Representations and Parametric Motion Fields,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 2238–2248.
- [10] E. Dupont, A. Goliński, M. Alizadeh, Y. W. Teh, and A. Doucet, “COIN: COmpression with Implicit Neural representations,” in *Proc. Int. Conf. Learn. Represent. Workshop*, 2021.
- [11] Y. Strümpfer, J. Postels, R. Yang, L. van Gool, and F. Tombari, “Implicit Neural Representations for Image Compression,” *arXiv:2112.04267*, Dec. 2021.
- [12] E. Dupont, H. Loya, M. Alizadeh, A. Goliński, Y. W. Teh, and A. Doucet, “COIN++: Neural Compression Across Modalities,” *arXiv:2201.12904*, June 2022.
- [13] J. Ramirez and J. Gallego-Posada, “L0onie: Compressing COINs with L0-constraints,” *arXiv:2207.04144*, July 2022.
- [14] M. Mancini, D. K. Jones, and M. Palombo, “Lossy compression of multidimensional medical images using sinusoidal activation networks: An evaluation study,” *arXiv:2208.01602*, Aug. 2022.
- [15] S. V. Venkatakrisnan, C. A. Bouman, and B. Wohlberg, “Plug-and-Play priors for model based reconstruction,” in *Proc. IEEE Glob. Conf. Signal Process. Inf. Process.*, 2013, pp. 945–948.
- [16] Y. Romano, M. Elad, and P. Milanfar, “The Little Engine That Could: Regularization by Denoising (RED),” *SIAM J. Imaging Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017.
- [17] J. Lee, J. Tack, N. Lee, and J. Shin, “Meta-learning sparse implicit neural representations,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 11769–11780.
- [18] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image segmentation using deep learning: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.
- [19] M. H. Hesamian, W. Jia, X. He, and P. Kennedy, “Deep learning techniques for medical image segmentation: Achievements and challenges,” *J. Digit. Imaging*, vol. 32, no. 4, pp. 582–596, 2019.
- [20] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, “nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation,” *Nat. Methods*, vol. 18, no. 2, pp. 203–211, 2021.
- [21] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.
- [22] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, “Implicit Neural Representations with Periodic Activation Functions,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 7462–7473.
- [23] M. Antonelli *et al.*, “The Medical Segmentation Decathlon,” *Nat. Commun.*, vol. 13, no. 1, pp. 4128, July 2022.
- [24] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv:1412.6980*, Jan. 2017.
- [25] C. Eldeniz *et al.*, “Phase2Phase: Respiratory Motion-Resolved Reconstruction of Free-Breathing Magnetic Resonance Imaging Using Deep Learning Without a Ground Truth for Improved Liver Imaging,” *Invest. Radiol.*, vol. 56, no. 12, pp. 809–819, 2021.