Patient-specific reconstruction of volumetric computed tomography images from a single projection view via deep learning

Liyue Shen1,2,3, Wei Zhao1,3 and Lei Xing1,2*

Tomographic imaging using penetrating waves generates cross-sectional views of the internal anatomy of a living subject. For artefact-free volumetric imaging, projection views from a large number of angular positions are required. Here we show that a deep-learning model trained to map projection radiographs of a patient to the corresponding 3D anatomy can subsequently generate volumetric tomographic X-ray images of the patient from a single projection view. We demonstrate the feasibility of the approach with upper-abdomen, lung, and head-and-neck computed tomography scans from three patients. Volumetric reconstruction via deep learning could be useful in image-guided interventional procedures such as radiation therapy and needle biopsy, and might help simplify the hardware of tomographic imaging systems.

The ability of computed tomography (CT) to take a deep and quantitative image of a patient or an object with high spatial resolution is highly valuable in scientific research and medical practice. Traditionally, a tomographic image is obtained via the mathematical inversion of the encoding function of the imaging wave for a given set of measured data from different angular positions (Fig. 1a,b). A prerequisite for artefact-free inversion is the satisfaction of the classical Shannon–Nyquist theorem in angular-data sampling, which imposes a practically achievable limit in imaging time and object irradiation. To mitigate the problem, image reconstruction with sparse sampling has been investigated extensively using techniques such as compressed sensing1–6 and maximum a posteriori7,8. These types of approaches introduce a regularization term to the inversion to encourage some ad hoc or presumed characteristics in the resultant image9–13. If imaging quality cannot be compromised, the resultant sparsity is generally limited and does not address the unmet demand for real-time imaging with substantially reduced subject irradiation (Fig. 1c). Indeed, while continuous efforts have been made to reduce the number of angular measurements in medical imaging, tomographic imaging with ultra-sparse sampling has not yet been realized.

In this study, we push sparse sampling to the limit of a single projection view and demonstrate single-view tomographic imaging with a patient-specific prior by leveraging deep learning and the seamless integration of prior knowledge in the data-driven image-reconstruction process. The harnessing of prior knowledge by machine-learning techniques in different data domains for improved imaging is an emerging topic of research. Some recent studies14–19 have also investigated machine-learning-based image reconstruction. Whereas the data-driven approach represents a potentially general strategy for image reconstruction, here single-view CT imaging is achieved via a patient-specific prior. Practically, it is actually advantageous to work with the patient-specific prior: for many image-guided interventional applications, the approach would enable scenarios most relevant to the specific patient under treatment.

Deep neural networks have attracted much attention for their ability to learn complex relationships and to incorporate existing knowledge into the inference model through feature extraction and representation learning19–22. The method has found widespread applications across disciplines, such as computer vision23–25, autonomous driving26, natural language processing27 and biomedical28–36. Here we design a hierarchical neural network for X-ray CT imaging with ultra-sparse projection views, and develop a structured training process for deep learning to generate three-dimensional (3D) CT images from two-dimensional (2D) X-ray projections. Our approach introduces a feature-space transformation between a 2D projection and a 3D volumetric CT image within a representation–generation (encoder–decoder) framework. By using the transformation module, we transfer the representations learned from the 2D projection to a representative tensor for 3D volume reconstruction in the subsequent generation network. Through the model-training process, the transformation module learns the underlying relationship between feature representations across dimensionality, making it possible to generate a volumetric CT image from a 2D projection. It should be emphasized that an X-ray projection is not a purely 2D cross-sectional image, as higher dimensional information is already encoded during the projection process (see schematic in Fig. 1a), with the encoding function determined by the physics of interactions between the X-ray and media. Generally, a single projection alone is not sufficient for capturing the anatomical information in the projection direction for the subsequent volumetric image reconstruction. What enables our deep-learning model for patient-specific volumetric image reconstruction is that anatomical relations (including the information in the direction of the projection view) are encoded during the model-training process via the use of augmented data-sets containing different 2D–3D data pairs of body positions and anatomical distributions. The deep-learning transformation deciphers the hidden information in the projection data and predicts a volumetric image with the help of prior knowledge gained during model training (Fig. 1d).
**Results**

Figure 2 shows the detailed structure of our deep-learning framework. The input to the neural network is single or multiple 2D projection images from different view angles. The output of the network is the corresponding volumetric CT image. During the model-training process, the neural network learns the mapping function from the 2D projection(s) to the volumetric image. Specifically, our deep-learning architecture consists of three main parts: a representation network, a transformation module and a generation network. The representation network extracts features from the input 2D projection data, the transformation module maps these features to a 3D representation, and the generation network reconstructs the volumetric image from the 3D features.
Embedding features and learns a semantic representation of the actual 3D scene from the input 2D projection(s). The transformation module bridges the representation and generation networks through convolution and deconvolution operations and relates the 2D and 3D feature representations. The role of the generation network is to provide volumetric images with subtle structures on the basis of the learned features from the representation network. In constructing the model, we assume that one or more 2D projections and the corresponding 3D image possess the same semantic representation, as they represent the same object or scene. In other words, the representation in feature space remains invariant in the transformation of a 2D projection into a 3D image. To a large extent, the task of 3D image reconstruction here is to train the encoder (that is, the representation network) and decoder (that is, the generation network) to reliably learn the relationship between the feature space and the image space. Details about the network architecture are included in Methods.

Training a deep-learning model requires a large amount of annotated data—this is often a bottleneck. Instead of actually measuring a large number of paired X-ray projections and CT images for supervised training, we digitally produce projection images from a CT image of a patient by using the geometry consistent with a clinical on-board cone-beam CT system for radiation therapy (Fig. 1a). For imaging in the thoracic or upper-abdominal region, where four-dimensional (4D) CT is often acquired to resolve organ motion caused by involuntary respiration, each 4D phase (that is, phase-resolved) CT is selected to form a 3D CT dataset. In reality, a 3D CT image captures only one out of numerous possible scenarios of the patient’s internal anatomy. To consider various clinical situations in the modelling, a series of translations, rotations and organ deformations are introduced to the 3D CT to mimic different imaging situations. For each of the transformations, the corresponding 2D projection image or digitally reconstructed radiograph (DRR) for one or more specified angles is produced. In this way, a dataset of DRR-CT pairs is generated for the training and testing of the deep-learning model. In practice, the dataset produced by using the CT of a given patient can be used to train a patient-specific deep-learning model for subsequent volumetric imaging of the same patient. The model can be used for interventional procedures such as radiation therapy and image-guided biopsy, in which pre-operative CT can be used to train the deep-learning model. One may, of course, construct a training dataset composed of an ensemble of patients with the above-mentioned DRR-CT pairs. This would lead to a more generally applicable model, but the fundamental principle would be the same. For simplicity, we will focus on the development of a patient-specific 2D–3D image mapping model.

We evaluate the approach by using different disease sites: an upper-abdominal case, a lung case and a head-and-neck case. We use the anterior–posterior 2D projection as input (Fig. 2). In all experiments, the same network architecture and training strategy are used. The loss curves (Fig. 3) indicate that the model is trained to fit the training data well and can also be generalized to work on the data not included in the training datasets. The details of dataset generation and the training process are described in Methods.

To evaluate the feasibility of the approach, we deploy the trained network on an independent testing dataset. Figure 4a shows our reconstruction results with a single anterior–posterior-view input for the abdominal CT and lung CT cases, together with the ground-truth CT images and the difference images between the obtained images and the ground truth. The deep-learning-derived images resemble the target images, indicating the potential of the model for volumetric imaging. We also reconstruct volumetric images...
with a single lateral view as input for the abdominal case, with similar results (see the Experiments section of the Supplementary Information). Furthermore, we use multiple quantitative evaluation metrics to measure the results. Table 1 summarizes the average values of the evaluation metrics. The qualitative and quantitative results demonstrate that our model is capable of achieving 3D image reconstruction even with only a single 2D projection. The results (Fig. 5) also confirm the validity of our approach.

In addition, we conduct experiments with two, five or ten projection views as inputs. The multiple-view angles are distributed evenly around a 180° semicircle (for instance, for two views, the two orthogonal directions are 0° (anterior–posterior) and 90° (lateral)). We stack the 2D projections from different view angles as the input data and modify the first convolution layer to fit the input channel size. With the same model-training procedure and hyper-parameters, we obtain the CT images for 2, 5 and 10 views for both the abdominal CT and lung CT cases (Fig. 4b–d). The quantitative evaluation of the results for these cases is summarized in Table 1. The training-loss curves and the corresponding coronal and sagittal views of the images are shown in Supplementary Figs. 1–6. For the abdominal CT case, 720, 180 and 600 images are used for training, validation and testing, respectively. For the lung CT case, 2,400, 600 and 200 images are used for training, validation and testing, respectively.

Discussion
To better understand the deep-learning model, we analyse the semantic representations learned from the model. Generally
speaking, successful generation of volumetric images is possible only if the model is able to learn the semantic representation of the 3D structure from the input projections. Thus, for the same volume, the representations obtained via learning from different angular projections should be similar, since they describe the same underlying 3D scene. In Fig. 6a, we visualize the feature maps extracted

Table 1 | Reconstruction results for the abdominal CT and lung CT cases

| Number of 2D projections | Abdominal CT |   | Lung CT |   |
|-------------------------|-------------|---|---------|---|
|                         | MAE | RMSE | SSIM | PSNR | MAE | RMSE | SSIM | PSNR |
| 1                       | 0.018 | 0.177 | 0.929 | 30.523 | 0.025 | 0.385 | 0.838 | 27.157 |
| 2                       | 0.015 | 0.140 | 0.945 | 32.554 | 0.024 | 0.399 | 0.837 | 26.985 |
| 5                       | 0.016 | 0.155 | 0.942 | 31.823 | 0.028 | 0.452 | 0.831 | 26.247 |
| 10                      | 0.018 | 0.165 | 0.939 | 31.355 | 0.027 | 0.429 | 0.817 | 26.636 |

MAE, mean absolute error; RMSE, root mean squared error; SSIM, structural similarity; PSNR, peak signal noise ratio.

Fig. 5 | Examples from the head-and-neck CT case. a, 3D CT images of a head-and-neck case used for the training of the deep-learning model. b, Left: testing samples and the corresponding difference images (with respect to the training samples) in the transverse, sagittal and coronal planes. Right: predicted images and the corresponding difference images (with respect to the ground truth) in the transverse, sagittal and coronal planes. For this case, 2,000, 500 and 200 images are used for training, validation and testing, respectively.
from the transformation module for two testing samples. For visualization purposes, only 5 randomly chosen channels among the 4,096 feature maps are shown, each with a size of $4 \times 4$ pixels. The feature maps learned from different numbers of 2D projections are displayed separately in different columns. The results show that, when different 2D views are given, the model extracts similar semantic representations of the underlying 3D scene. Furthermore, Fig. 6b shows the visualization of $t$-distributed stochastic neighbour embedding ($t$-SNE) for the feature maps of 15 testing samples. The $t$-SNE technique is commonly used to visualize high-dimensional data by embedding each sample as a point in a 2D space. The four points in a cluster represent the learned features from one-, two-, five- and ten-view reconstruction models. Each cluster denotes the embedded representations for each of the 15 randomly chosen testing samples.

We also measure the similarity of the embedding representations by calculating the Euclidean distance between two feature maps. In this way, we compute a similarity score, ranging from 0 to 1, where high similarity (a score approaching 1) indicates that the distance between two feature maps is close to zero. We plot a correlation matrix (Fig. 6c) among 50 randomly selected testing samples, with their feature representations extracted from one-view and two-view reconstruction models. The highest values stand out in the diagonal of the correlation matrix whereas other off-diagonal values remain relatively low. This illustrates that the two sets of feature representations learned from one-view and two-view projections for the same 3D scene are more similar or closer in Euclidean distance space compared with the feature representations learned from other different 3D scenes. This provides additional evidence supporting the capability of the model to learn a semantic representation of the 3D scene with a single projection.

Robustness against possible irregular breathing patterns is important for future clinical implementation of the approach. The robustness of deep networks against various perturbations is an intense area of research in artificial intelligence. As summarized in ref. 43, possible solutions come in three categories: (1) the modification of network architectures (for example, adding more layers, changing the loss function and modifying the activation functions); (2) the use of external models as a network add-on to detect out-of-distribution data (for example, using an external detector to rectify the irregular data); and (3) the modification of the training-data distribution or the training strategy (for example, adding regularization, data augmentation or leveraging adversarial training). In (1), the efforts are focused on refining the learning models. In (2), irregular motions might be regarded as out-of-distribution data, where some potential techniques, such as a detector subnetwork or the confidence-based method, might be useful for detecting irregular input. Among the various methods, the modification of the training-data distribution is arguably the most straightforward way.
to proceed. The rationale is that if the irregularities can be incorporated effectively into the training dataset and the training strategy can be adjusted accordingly, the robustness of the trained model would be enhanced. To a certain extent, this has been elaborated in the example in Supplementary Fig. 7, where it is demonstrated that, because of the inclusion of augmented training datasets with rotational transformations, the deep-learning approach is much more robust against a small rotation of the imaging subject than a conventional principal component analysis (PCA)-based method. Quantitative results of the study for the testing sample presented in Supplementary Fig. 7 are shown in Supplementary Table 1.

**Outlook.** We have described a deep-learning approach for volumetric imaging with ultra-sparse data sampling and a patient-specific prior. The data-driven strategy is capable of holistically extracting the feature characteristics embedded in a single projection or in a few 2D projections, and of transforming them into the corresponding 3D image through model learning. The image-feature space transformation plays an essential role in the ultra-sparse image reconstruction. At the training stage, the method incorporates diverse forms of a priori knowledge into the reconstruction. The manifold-mapping function is learned from the training datasets, rather than relying on any ad hoc form of motion trajectory. Although we have used X-ray imaging and patient-specific data, the concept and implementation of the approach could be extended to other imaging modalities or to other data domains with ultra-sparse sampling. Practically, single-view imaging represents a potential solution for many image-guided interventional procedures and may help to simplify the hardware of tomographic imaging systems.

**Methods**

**Problem formulation.** We formulate the problem of 3D image reconstruction from 2D projection(s) into a deep-learning framework. Given a sequence of 2D projections denoted as \( \{X_1, X_2, \ldots, X_N\} \), where \( X_i \in \mathbb{R}^{m \times n \times W_0} \) for all \( 1 \leq i \leq N \) and \( N \) is the number of given 2D projections, the goal is to generate a volumetric 3D image \( T \) describing the corresponding 3D physical scene. With the sequence of 2D projections as input, the deep-learning model outputs the predicted 3D volume denoted as \( Y_{\text{pred}} \in \mathbb{R}^{m \times n \times W_0} \), while \( Y_{\text{truth}} \in \mathbb{R}^{m \times n \times W_0} \) is the ground-truth 3D image as the reconstruction target. Note that network prediction \( Y_{\text{pred}} \) is of the same size as ground-truth image \( Y_{\text{truth}} \), where each entry is a voxel-wise intensity value. Thus, the problem is formulated as finding a mapping function \( F \) transforming 2D projections to volumetric images. To tackle this problem, a deep-learning model is trained to find the mapping function \( F \), which uses 2D projections \( \{X_1, X_2, \ldots, X_N\} \) as input and predicts the corresponding 3D image \( Y_{\text{pred}} \), as expressed in Equation (1).

\[
F(X_1, X_2, \ldots, X_N) = Y_{\text{pred}}
\]  

(1)

In order to use a sequence of 2D projections as model input, we stack all the 2D projections together as a single 3D tensor. In other words, a set of 2D projections \( \{X_1, X_2, \ldots, X_N\} \) \( X_i \in \mathbb{R}^{m \times n \times W_0} \) are stacked as a 3D volume \( \mathbf{X} \in \mathbb{R}^{n \times m \times W_0} \), where \( n \) is the number of 2D projections. In what follows, we introduce the model architecture of the deep neural network in detail.

**Encoder–decoder framework.** The deep neural network is formulated into an encoder–decoder framework (Fig. 3). In the auto-encoder model, the encoder converts high-dimensional data into embedded representations whereas the decoder reconstructs high-dimensional input. In our task, instead of decoding to get the input, we developed a modified decoder to generate the corresponding volumetric images based on the codes converted by the encoder. More precisely, with a sequence of 2D projections as input, the encoder network learns the feature representation by extracting semantic information from 2D projections, such as organ position and size. In this way, the encoder network learns a transformation function \( h_{\text{L}} \), from the 3D image domain to the feature domain. A transformation module then follows to learn the manifold-mapping function \( h_{\text{R}} \) in the feature domain to transform the feature representation across dimensions. Using the learned feature representation as the input, the decoder network is trained to generate the 3D volume. In other words, the decoder network learns a transformation function \( h_{\text{R}} \), from the feature domain to the 3D image domain. In this way, we fit the target mapping function \( F \) by decomposition: \( F = h_{\text{R}} \circ h_{\text{L}} \).

The rationale behind our network design is that both 2D projections and 3D images share the same semantic feature representation in the feature domain, as they represent the image expressions of the same object or physical scene. Accordingly, the representation in feature space should remain invariant. In a sense, if the model can learn the transformation function between the feature space and the 2D and/or 3D image space, it is possible to reconstruct 3D images from 2D projections. Therefore, following this encoder–decoder framework, our model is able to learn how to generate 3D images from 2D projections by using the learned representations in high-dimensional feature space.

**Representation network.** Superb performance has been achieved by deep residual networks11 such as ResNet12 in many tasks. A key step in residual learning is the identity mapping that facilitates the training process and avoids gradient vanish in back-propagation11, which encourages residual learning of the hierarchical representation at each stage and eases the training of the deep network. Motivated by this feature, we introduce a residual-learning scheme in the representation network (Fig. 2), in which the 2D convolution residual block is used to assist the deep-learning model to learn semantic features from 2D projections. Details about the residual-learning scheme are presented in Supplementary Information, Ablative study and discussion, and the results are summarized in Supplementary Table 2. Specifically, each 2D convolution residual block consists of a pattern of 2D convolution layer (with kernel size 4 and stride 2) \( \rightarrow \) 2D batch normalization layer \( \rightarrow \) ReLU layer \( \rightarrow \) 2D convolution layer (with kernel size 3 and stride 1) \( \rightarrow \) 2D batch normalization layer \( \rightarrow \) ReLU layer. The first layer performs 2D convolution operations using a \( 4 \times 4 \) kernel with sliding stride \( 2 \times 2 \), which down-samples the spatial size of the feature map by a factor 2. In addition, to keep the sparsity of high-dimensional feature representation, we correspondingly double the channel number of the feature maps by increasing the number of convolutional filters in each convolution layer, and the ReLU activation and batch normalization5 then follows before feeding the feature maps through the ReLU layer11. Next, the second 2D convolution layer and 2D batch normalization layer are done by a kernel size of \( 3 \times 3 \) and sliding stride \( 1 \times 1 \), which keeps the spatial shape of the feature maps. Moreover, before applying the second ReLU layer, an extra shortcut path is established to add up the output of the transformation module to help the knowledge transfer through this module. By setting up the shortcut path of identity mapping, the second convolution layer is encouraged to learn the residual feature representations. To extract hierarchical semantic features from 2D projections, we constructed the representation network by concatenating five 2D convolution residual blocks with different number of convolutional filters. A detailed discussion of the network depth is available in Supplementary Information, Ablative study and discussion, with some results illustrated in Supplementary Fig. 8. To be concise, we use the notation \( k \times n \times x \) to denote \( k \) channels of feature maps in a spatial size of \( m \times n \). In the generation network, the size of input images is denoted as \( N \times 128 \times 128 \), where \( N \) is the number of 2D projections. The change of feature-map size through the network is \( N \times 128 \times 128 \rightarrow 256 \times 64 \times 64 \rightarrow 512 \times 32 \times 32 \rightarrow 1024 \times 16 \times 16 \rightarrow 2048 \times 8 \times 8 \rightarrow 4096 \times 4 \times 4 \), where each ‘\( \rightarrow \)’ represents going through a 2D convolution residual block as described above, except that batch normalization and ReLU activation are removed in the first convolution layer. Thus, the output of the representation network is a feature representation extracted from 2D projections with a size of \( 4096 \times 4 \times 4 \).

**Transformation module.** To bridge the representation and generation networks, a transformation module is deployed after learning the representations. As shown in Fig. 5, by taking the convolution operations with a kernel size of \( 1 \times 1 \) and ReLU activation, the 2D convolution layer learns a transformation across all 2D feature maps. Then, we reshape embedded representations from \( 4096 \times 4 \times 4 \) to \( 2048 \times 2 \times 2 \times 2 \times 2 \). In this way, we transform the feature representation across dimensions for subsequent 3D volume generation. Next, a 3D deconvolution layer with a kernel size of \( 1 \times 1 \times 1 \) and sliding stride of \( 1 \times 1 \times 1 \) learns a transformation among all 3D feature cubes while keeping the feature size unchanged. This transformation module bridges the 2D and 3D feature space. Moreover, as described in previous work11, we also remove the batch normalization in the transformation module to help the knowledge transfer through this module.

**Generation network.** The generation network is built upon the 3D deconvolution block, which consists of a pattern of 3D deconvolution layer (with kernel size 4 and stride 2) \( \rightarrow \) 3D batch normalization layer \( \rightarrow \) ReLU layer \( \rightarrow \) 3D deconvolution layer (with kernel size 3 and stride 1) \( \rightarrow \) 3D batch normalization layer \( \rightarrow \) ReLU layer. Note that the ‘deconvolution’ layer actually means the operation of ‘transformed convolution’ or fractional stride convolution which performs up-sampling operation. The first deconvolution layer up-samples a feature map by a factor 2 with a \( 4 \times 4 \times 4 \) kernel and sliding stride \( 2 \times 2 \times 2 \). To transform from a high-dimensional feature domain to a 3D image domain, we accordingly reduce the number of feature maps by decreasing the number of deconvolutional filters. The second deconvolution layer with a \( 3 \times 3 \times 3 \) kernel and sliding stride \( 1 \times 1 \times 1 \) keeps the spatial shape of feature maps. A 3D batch normalization layer and a ReLU layer follow after each deconvolution layer. Hierarchically, the 3D generation network consists of five concatenated deconvolutional blocks. Following the same convention as in the representation network, we use a notation of \( k \times m \times n \times x \) to denote \( k \) channels of 3D feature maps with a spatial size of \( m \times n \times x \). With a representation input of \( 2048 \times 4 \times 4 \times 4 \), the data flow of the feature maps is \( 2048 \times 2 \times 2 \times 2 \rightarrow 1,048 \times 4 \times 4 \times 4 \rightarrow 512 \times 8 \times 8 \times 8 \rightarrow 256 \times 16 \times 16 \times 16 \rightarrow 128 \times 32 \times 32 \times 32 \rightarrow 64 \times 64 \times 64 \times 64 \).
CT images are acquired with 120 kV, 80 mA on a positron emission tomography–DRR samples independently from the remaining 4 phases of the 4D CT. The 4D patient anatomy for model training and 180 DRRs for validation. To ensure that are used to generate the CT-DRR pairs for model training and validation with sites. In the first study, a ten-phase upper-abdominal 4D CT scan of a patient for Materials.

is no ReLU layer after the final convolution layer.

2D projection images are resized to 128

network. First, we resize all data samples to the same size. For example, all the Image pre-processing. TrueBeam system. To build a reliable model, the training and testing datasets Varian Real-time Position Management system (Varian Medical Systems). The CT simulator (Biograph mCT 128, Siemens Medical Solutions) together with a 2D axis. Each data sample is a pair

of 2D projected view(s) and the corresponding 3D CT. Similar to other deep-learning-based method, small motions of less than 3 mm may not be improved as computational technology advances. In practice, methods such as

be improved as computational technology advances. In practice, methods such as deep-learning-based super-resolution are being actively pursued, which may be employed to improve the spatial resolution of the approach. Additionally, following the standard protocol of data pre-processing, we conduct scaling normalization for both the 2D projections and the 3D volumetric images, where pixel-wise or voxel-wise intensities are normalized to the interval [0,1]. Moreover, we normalize the statistical mean and standard derivation among all the training data. When a new sample is inputted, we subtract the mean value from the input image(s) and divide the image(s) by the standard derivation to get the input 2D image(s).

Training details. With input images X containing a stacked sequence of 2D projections, we train the deep network to predict the volumetric images $Y_{\text{truth}}$, which is expected to be as close as possible to the ground-truth images $Y_{\text{truth}}$. We define the cost function as the MSE between the prediction $Y_{\text{pred}}$ and the ground truth $Y_{\text{truth}}$ and the model was optimized by stochastic gradient descent iteratively. For comparison, we used the same training strategy and hyper-parameters for all experiments. We implemented the network by using the PyTorch library, and used the Adam optimizer to minimize the loss function and to update the network parameters iteratively through back-propagation. A learning rate of 0.00002 and a mini-batch size of 1 are used because of memory limitations. At the end of each training epoch, the model is evaluated on the validation set. This strategy is commonly used to monitor the model performance and avoid overfitting the training data. In addition, the learning rate is scheduled to decay according to the validation loss. Specifically, if the validation loss remains unchanged for 10 epochs, the learning rate is reduced by a factor 2. Finally, the best checkpoint model with the smallest validation loss is saved as the final model in the experiments. We trained the network using one Nvidia Tesla V100 graphics processing unit for 100 epochs (duration typically around 20 h for the abdominal CT case). During testing, the typical inference time for 3D reconstruction of one testing sample is around 0.5 s.

Evaluation. To evaluate the performance of the approach, we deploy the trained model on a testing dataset, and analyze the reconstruction results using both qualitative and quantitative evaluation metrics. We use four different metrics to measure the quality of predicted 3D images: MAE, RMSE, SSIM and PSNR.

We compute the average values across all testing samples, and they are shown in Table 1. MAE/MSE is the L1-norm/L2-norm error between $Y_{\text{pred}}$ and $Y_{\text{truth}}$. As usual, we take the square root of MSE to get RMSE. In practice, MAE and RMSE are commonly used to estimate the difference between the prediction and ground-truth images. SSIM score is calculated with a windowing approach in an image, and is used to measure the overall similarity between two images. In general, a lower value of MAE and RMSE or a higher SSIM score indicates a better prediction closer to the ground-truth images. PSNR is defined as the ratio between the maximum signal power and the noise power that affects the image quality. PSNR is widely used to measure the quality of image reconstruction.

Comparison study. To better benchmark the proposed method against the existing techniques, we conduct a comparative study with the published PCA-based method and elaborate the difference and advantages of our proposed approach. The comparison is done for a special situation of 4D CT reconstruction (abdominal CT), where the anatomical motion may be characterized by principal components. We find that the PCA and deep-learning-based methods produce similar results in an ideal case when there is no inter-scan variation in patient positioning (since the results are very similar, the resultant images are not shown). However, the deep-learning model outperforms the PCA method in more realistic scenarios when the patient position deviates slightly from that of the reference scan (see Supplementary Information for details).

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

The authors declare that the main data supporting the results in this study are available within the paper and its Supplementary Information. The raw datasets from Stanford Hospital are protected because of patient privacy yet can be made available upon request provided that approval is obtained after an Institutional Review Board procedure at Stanford.

Code availability

The source code of the deep-learning algorithm is available for research uses at https://github.com/liyues/PatRecon. The source code is available within the paper and its Supplementary Information. The raw datasets from Stanford Hospital are protected because of patient privacy yet can be made available upon request provided that approval is obtained after an Institutional Review Board procedure at Stanford.

Received: 29 November 2018; Accepted: 19 September 2019; Published online: 28 October 2019

References

1. Candes, E. J., Romberg, J. K. & Tao, T. Stable signal recovery from incomplete and inaccurate measurements. Commun. Pure Appl. Math. 59, 1207–1223 (2006).
2. Lustig, M., Donoho, D. & Pauly, J. M. Sparse MRI: The application of compressed sensing for rapid MR imaging. Magn. Reson. Med. 58, 1182–1195 (2007).
3. Sidky, E. Y. & Pan, X. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. Phys. Med. Biol. 53, 4777–4807 (2008).
4. Chen, G. H., Tang, J. & Leng, S. Prior image constrained compressed sensing (B-SCCS): a method to reconstruct dynamic CT images from highly undersampled projection data sets. Med. Phys. 35, 660–663 (2008).
5. Yu, H. & Wang, G. Compressed sensing based interior tomography. Phys. Med. Biol. 54, 2791–2805 (2009).
6. Choi, K., Wang, J., Zhi, L., Suh, T.S., Boyd, S. & Xing, L. Compressed sensing based cone-beam computed tomography reconstruction with a first-order method. Med. Phys. 37, 5113–5125 (2010).
7. Fessler, J. A. & Rogers, W. L. Spatial resolution properties of penalized-likelihood image reconstruction: space-invariant tomographs. IEEE Trans. Image Process 5, 1346–1358 (1996).
8. Jt, S., Xue, Y. & Carin, L. Bayesian compressive sensing. IEEE Trans. Signal Process. 56, 2346–2356 (2008).
9. Engl, H. W., Hanke, M. & Neubauer, A. Regularization of inverse problems, Vol. 375 (Springer Science & Business Media, 1996).
10. Stayman, J. W. & Fessler, J. A. Regularization for uniform spatial resolution properties in penalized-likelihood image reconstruction. IEEE Trans. Med. Imaging 19, 601–615 (2000).
39. Papernot, N., McDaniel, P. & Goodfellow, I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *J. Mach. Learn. Res.*, 46, 3142–3155 (2019).

40. Eslami, S. A. et al. Neural scene representation and rendering. *Neural Netw.*, 61, 85–117 (2015).

41. Xu, Q. et al. Low-dose X-ray CT reconstruction by domain-transform manifold learning. *Nature*, 555, 487–492 (2018).

42. Shen, L., Yeung, S., Hoffman, J., Mori, G. & Fei-Fei, L. Scaling Human-Object Affordance for Direct Perception in Autonomous Driving. In *Proc. 31st Conference on Advances in Neural Information Processing Systems* (eds Bengio, S. et al.) 160–167 (IEEE, 2018).

43. Xu, Q., Xu, Q., Shirato, H., Shimizu, S. & Xing, L. Projection-domain scatter correction for cone beam computed tomography using a residual convolutional neural network. *Med. Phys.*, 46, 3142–3155 (2019).

44. Wu, Y. et al. Incorporating prior knowledge via volumetric deep residual network to optimize the reconstruction of sparsely sampled MRI. *Magn. Reson. Imaging* https://doi.org/10.1016/j.mri.2019.03.012 (2019).

45. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *ICLR*, 2015.

46. Ioffe, S. & Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proc. 32nd International Conference on Machine Learning* (ICML, 2015).

47. Li, R. et al. 3D tumor localization through real-time volumetric x-ray imaging: a comparison study of 2D convolutional networks and 3D networks. *IEEE Trans. Med. Imaging*, 38, 2211–2223 (2017).

48. Goyal, P., Dollár, P., Danelljan, M. & Fauw, M. Context-aware semantic segmentation. In *Proc. 32nd Conference on Neural Information Processing Systems* (ICLR, 2018).

49. Akhtar, N. & Mian, A. Threat of adversarial attacks on deep learning in computer vision: a survey. *IEEE Access*, 6, 14410–14430 (2018).

50. Esteva, A. et al. Dermatologist-level classification of skin cancer with deep learning. *IEEE Trans. Neural Netw. Learn. Syst.*, 30, 2805–2824 (2019).

51. Hinton, G. E. & Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science*, 313, 504–507 (2006).

52. Ike, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2016).

53. Nair, V. & Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proc. 27th International Conference on Machine Learning* 807–814 (ICML, 2010).

54. Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A. Image-to-image translation with conditional adversarial networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 1125–1134 (IEEE, 2017).

55. Paszke, A. et al. Automatic differentiation in pytorch. In *Proc. 30th Conference on Advances in Neural Information Processing Systems* (NIPS, 2017).

56. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. In *Proc. 3rd International Conference on Learning Representations* (ICLR, 2015).

57. Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process*, 13, 600–612 (2004).

58. Li, R. et al. Real-time volumetric image reconstruction and 3D tumor localization based on a single x-ray projection image for lung cancer radiotherapy. *Med. Phys.*, 37, 2822–2826 (2010).

59. Li, R. et al. 3D tumor localization through real-time volumetric x-ray imaging for lung cancer radiotherapy. *Med. Phys.*, 38, 2783–2794 (2011).

60. Xu, Y. et al. A method for volumetric imaging in radiotherapy using single x-ray projection. *Med. Phys.*, 42, 2498–2509 (2015).

Acknowledgements

This research is partially supported by the National Institutes of Health (R01CA176553 and R01EB016777). The contents of this article are solely the responsibility of the authors and do not necessarily represent the official NIH views.

Author contributions

L.X. proposed the original notion of single-view reconstruction for tomographic imaging and supervised the research. L.S. designed and implemented the algorithm. W.Z. designed the experiments and implemented the data generation process. L.S. and W.Z. carried out experimental work. L.X., L.S. and W.Z. wrote the manuscript. All the authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41551-019-0466-4.

Correspondence and requests for materials should be addressed to L.X.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature Limited 2019
Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see Authors & Referees and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

- The exact sample size \( (n) \) for each experimental group/condition, given as a discrete number and unit of measurement
- A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
- The statistical test(s) used AND whether they are one- or two-sided
- A description of all covariates tested
- A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
- A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
- For null hypothesis testing, the test statistic (e.g. \( F, t, r \)) with confidence intervals, effect sizes, degrees of freedom and \( P \) value noted

Give \( P \) values as exact values whenever suitable.

For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings

For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes

Estimates of effect sizes (e.g. Cohen's \( d \), Pearson's \( r \)), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection

All patient data were acquired with 120 kV on a Positron emission tomography–computed tomography (PET-CT) simulator (Biograph mCT 128, Siemens Medical Solutions, Erlangen, Germany).

Data analysis

Python (version 3.5.5), PyTorch (version 0.4.1), and custom algorithms.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

The authors declare that the main data supporting the results in this study are available within the paper and its Supplementary Information. The raw datasets from Stanford Hospital are protected because of patient privacy yet can be made available upon request provided that approval is obtained after an Institutional Review Board procedure at Stanford.
Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☑ Life sciences  ☐ Behavioural & social sciences  ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

- **Sample size**
  - For abdominal CT, 720, 180, and 600 images are used for training, validation, and testing. For lung CT, 2400, 600, and 200 images are used for training, validation, and testing. For head and neck CT, 2000, 500, and 200 images are used for training, validation, and testing.

- **Data exclusions**
  - No data were excluded from the analyses.

- **Replication**
  - All experiments were conducted in a manner that produced clean replicates. The algorithm performance should have similar results if the model is trained following the training strategy described in the paper.

- **Randomization**
  - The training and validation datasets were randomly selected.

- **Blinding**
  - The test datasets were different from the training and testing datasets.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

| Materials & experimental systems | Methods |
|----------------------------------|---------|
| n/a | n/a |
| ☑ | ☑ |
| ☐ | ☐ |
| ☐ | ☐ |
| ☐ | ☐ |
| ☐ | ☐ |
| ☐ | ☐ |
| ☒ | ☒ |
| ☐ | ☐ |
| ☐ | ☐ |
| ☐ | ☐ |
| ☐ | ☐ |
| ☐ | ☐ |
| ☐ | ☐ |
| ☐ | ☐ |
| ☐ | ☐ |
| ☐ | ☐ |
| ☐ | ☐ |

Clinical data

Policy information about clinical studies

All manuscripts should comply with the ICMJE guidelines for publication of clinical research and a completed CONSORT checklist must be included with all submissions.

- **Clinical trial registration**
  - The study is not a clinical trial.

- **Study protocol**
  - The study is not a clinical trial.

- **Data collection**
  - All patient data were acquired with 120 kV on a Positron emission tomography–computed tomography (PET-CT) simulator (Biograph mCT 128, Siemens Medical Solutions, Erlangen, Germany).

- **Outcomes**
  - The study is not a clinical trial.