Learning a Generative Motion Model from Image Sequences based on a Latent Motion Matrix

Julian Krebs, Hervé Delingette, Nicholas Ayache, Tommaso Mansi

Learning a Generative Motion Model from Image Sequences based on a Latent Motion Matrix

Julian Krebs, Hervé Delingette, Nicholas Ayache and Tommaso Mansi

***Abstract*—We propose to learn a probabilistic motion model from a sequence of images for spatio-temporal registration. Our model encodes motion in a low-dimensional probabilistic space – the motion matrix – which enables various motion analysis tasks such as simulation and interpolation of realistic motion patterns allowing for faster data acquisition and data augmentation. More precisely, the motion matrix allows to transport the recovered motion from one subject to another simulating for example a pathological motion in a healthy subject without the need for inter-subject registration. The method is based on a conditional latent variable model that is trained using amortized variational inference. This unsupervised generative model follows a novel multivariate Gaussian process prior and is applied within a temporal convolutional network which leads to a diffeomorphic motion model. Temporal consistency and generalizability is further improved by applying a temporal dropout training scheme. Applied to cardiac cine-MRI sequences, we show improved registration accuracy and spatio-temporally smoother deformations compared to three state-of-the-art registration algorithms. Besides, we demonstrate the model's applicability for motion analysis, simulation and super-resolution by an improved motion reconstruction from sequences with missing frames compared to linear and cubic interpolation.**

***Index Terms*—motion model, deformable registration, conditional variational autoencoder, gaussian process, latent variable model, motion interpolation, motion simulation, tracking.**

I. Introduction

MOTION analysis is an important task in many medical image analysis problems such as organ tracking or longitudinal analysis of various diseases. For moving organs such as the heart, it is not only important to track anatomical structures but also to analyze motion indices that are useful for disease diagnosis or therapy selection [2]. Extracting motion patterns further allows to compensate for motion, handle missing data or do temporal super-resolution and motion simulation.

Motion in medical image sequences is typically analyzed by computing temporally consistent pairwise deformations where each frame in a sequence is registered to a target frame [2]. The resulting series of deformation fields can be utilized to track structures throughout the sequence and to identify abnormal motion patterns, for example by computing clinically relevant variables such as the ejection fraction (EF) of the heart [3].

This work has been supported by the French government, through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002 and the grant AAP Santé 06 2017-260 DGA-DSH, and by the Inria Sophia Antipolis - Méditerranée, "NEF" computation cluster. The used data were obtained from the EU FP7-funded project MD-Paedigree and the ACDC STACOM challenge 2017 [1].

J. Krebs is with the Université Côte d'Azur, Inria, Epione Team, Sophia Antipolis, 06902 France and also Siemens Healthineers, Digital Technology & Innovation, Princeton, NJ 08540 USA (e-mail: julian.krebs@inria.fr).

H. Delingette and N. Ayache are with the Université Côte d'Azur, Inria, Epione Team, Sophia Antipolis, 06902 France.

T. Mansi is with Siemens Healthineers, Digital Technology & Innovation, Princeton, NJ 08540 USA.

A. State-of-the-art

Registration algorithms typically seek to find the deformation field between two images by solving an optimization problem consisting of a similarity metric and a regularizer. The similarity metric measures the distance between the two images while the regularizer constrains the smoothness of the resulting deformation field. A large variety of registration algorithms using different similarity and regularizing metrics have been proposed [4]. One group of registration methods aims to ensure diffeomorphic deformations due to their favorable properties. Diffeomorphisms are topology-preserving and invertible deformations which makes them suitable for many medical registration problems in which foldings are physically implausible [5]. This makes diffeomporphisms also appropriate for tracking anatomical structures in image sequences such as in cardiac imaging [6] (assuming structures do not go out of the field of view). Many diffeomorphic registration algorithms have been proposed such as [5], [7]–[9], the SyN algorithm [10] and the LCC-demons [11]. Recently, learning-based algorithms for pairwise diffeomorphic registration have been proposed. These are based on supervised *ground-truth* deformations [12], [13] or on unsupervised learning [14], [15]. The latter are trained by minimizing a loss function consisting of an image similarity and a deformation regularizer, similarly to the traditional optimization problem, which has been proposed earlier for learning-based non-diffeomorphic registration [16], [17]. In [14], [15], diffeomorphisms are guaranteed by using the stationary velocity field (SVF) parameterization based on the scaling-squaring algorithm [18].

For image sequences, one difficulty is to acquire temporally smooth deformations that are fundamental for consistent tracking. That is why registration algorithms with a temporal regularizer have been proposed [19]–[24]. For respiratory motion modeling, learning-based regression models using ultrasound and MR images of different respiratory states have been used to learn respiratory motion patterns [25], [26]. In the computer vision community, temporal video super-resolution and motion compensation are a related research topic [27], [28].

However, while these methods are able to capture temporally consistent deformations along a sequence of images, they do not extract intrinsic low-dimensional motion parameters crucial for building a comprehensive motion model that can

be used for analysis tasks such as motion simulation, transport or classification as it is for example done in bio-mechanical models such as [29]. Yang et al. [30] generated a motion prior using manifold learning from low-dimensional shapes. Qiu et al. [31] proposed to build an eigenspace of initial momenta using PCA. In an image-driven fashion, Rohé et al. [3] introduced a parameterization, the Barycentric Subspaces, for cardiac motion analysis. It has been also proposed to learn statistical models for respiratoy motion by combining sample transformations derived from shapes [32] or images directly [33]. However, these models require either to extract shapes from images and/or an inter-subject alignment which can be a difficult task with regard to different use cases.

B. Learning a Probabilistic Motion Model

In contrast, we propose a probabilistic motion model that is built in a fully data-driven way from image sequences. Our model learns a low-dimensional motion matrix in an unsupervised fashion from sequences with constant intensity levels. Instead of defining a motion parameterization explicitly or learning from pre-processed shapes or pre-aligned sequences, an application-specific motion model is learned. The advantage of such an application-specific model is that it is guided from training images which allows to unveil inherent characteristic motion features instead of for example relying on pre-defined bio-mechanical parameters often based on limiting assumptions. The goal is not only to retrieve a compact representation of the motion but to obtain a structured and generative encoding that allows for temporal interpolation (to predict missing frames) and to simulate an indefinite number of new motion patterns. These features could be helpful for data augmentation and to speed-up image acquisition as the model reconstructs a full cyclic motion from missing frames. As for all learning-based approaches, our resulting model is biased on the training data which mostly impacts its generative abilities. Naturally, it will tend to simulate pathological cases if trained mostly on image sequences containing similar pathologies. Besides, the application-specific probabilistic encoding could be useful for group-wise analysis as it enables to transport motion characteristics to a new subject – without requiring inter-subject alignment – simulating for example a pathological motion in a healthy subject. This can be useful for data augmentation and class balancing for instance by generating many simulated examples of a certain disease.

In this work, we introduce a novel Gaussian process (GP) prior to extend a conditional variational autoencoder (CVAE [34]), a latent variable model, for temporal sequences. The GP prior constrains the standard independence assumption between all variables of CVAEs with a temporal regularizer. Relating latent variables over time leads to higher temporal consistency that can improve the tracking of for example moving organs. A pairwise encoder-decoder neural network applies a temporal convolutional network (TCN) in its latent space in order to learn intrinsic temporal dependencies. Furthermore, we utilize a self-supervised training scheme based on temporal dropout (TD) to enforce temporal consistency and increase generalizability of the motion model. Smooth and diffeomorphic deformations are guaranteed by applying an exponentiation layer [15] and spatio-temporal regularization.

The proposed model demonstrates state-of-the-art registration accuracy measured on segmentation overlaps and distances and regularity for diffeomorphic tracking of cardiac cine-MRI. In addition, the potentials of the generated latent motion matrix for motion simulation, interpolation and transport are demonstrated. The main contributions are as follows:

- An unsupervised probabilistic motion model learned from medical image sequences
- A conditional VAE model trained with a novel Gaussian process prior and self-supervised temporal dropout using temporal convolutional networks
- Demonstration of cardiac motion tracking, simulation, transport and temporal super-resolution

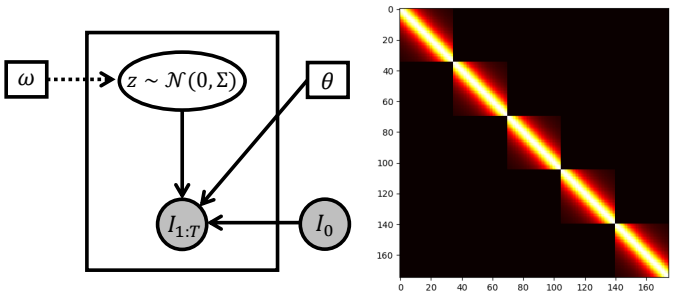
This paper extends our preliminary conference paper [35] by replacing the standard unit Gaussian of the CVAE with a novel Gaussian process prior. We add detailed derivations of the motion model and show improved tracking accuracy and temporal smoothness. Finally, we show a first generalization of the model to 3D+t sequences.

II. Methods

In this work, motion is described by deformation fields between one reference image, for example the first frame, and all other images in an image sequence $I_{0:T}$ with $T+1$ frames. In order to extract consistent sequential deformations ϕ_t with $t \in [1, T]$, we propose a temporal latent variable model that encodes the motion in a low-dimensional probabilistic space, the motion matrix $z \in \mathbb{R}^{D \times T}$.Here, we define the reference image I_0 as moving image, while the other frames are fixed images I_t. Each image pair (I_0, I_t) is encoded by D latent variables, the $z_{\cdot t}$-code, which are the columns of z. Each $z_{\cdot t}$ parameterizes the deformation field ϕ_t while being conditioned on the moving image I_0. The rows $z_{d\cdot}$ with length T of the motion matrix z represent the encoded deformation sequence per latent dimension $d \in D$.

Our motion model is learned from data by imposing a Normal prior distribution $p(z)$ on the latent variables z that follows a Gaussian process (GP) prior in the temporal dimension for each $z_{d\cdot}$. In addition, we assume independence between the latent variables $z_{d\cdot}$ as in standard VAEs [36]. Note, when z is written as part of a distribution like $p(z)$, z is used as a vector of size DT in row-major order rather than a matrix for simpler notation.

During training, we follow the learning paradigms of conditional variational autoencoders (CVAE [34], [37]) with the exception of replacing the multivariate unit Gaussian prior with the proposed GP-prior. The approximated posterior is the output of a temporal convolutional neural network (TCN [38]) allowing for temporal regularization. To further facilitate temporal dependencies and handle missing data, temporal dropout (TD) is applied during the training procedure. In the following, the different parts of the method are explained. First, the probabilistic motion model using a GP-prior is defined. Then, posterior and data likelihood distributions are modeled using a encoder-decoder neural network. Lastly, the concept of temporal dropout is introduced.



(a) (b)

Fig. 1. (a) Generative process for the motion model representing the likelihood of fixed images $I_{1:T}$ given the latent variables z and moving image I_0: $p_\theta(I_{1:T}|z, I_0)$, where ω and θ are fixed parameters and arrows denote dependencies between random variables. (b) Visualization of the covariance matrix Σ of the Gaussian prior $p(z)$ with 5 latent dimensions, a sequence time length of 35 and a length scale of the Cauchy kernel of 7.

A. Generative Motion Model using a Gaussian Process Prior

The proposed motion model consists of an encoder $q_\omega(z|I_{0:T})$ and a decoder $p_\theta(I_{1:T}|z, I_0)$ which are parameterized by ω and θ respectively. The encoder first independently maps each image pair (I_0, I_t) to a pair-wise latent representation which is then temporally regularized by mixing all time steps using multiple temporal 1D convolutions (TCN) to retrieve a joint latent representation (μ, σ). The motion matrix z is finally extracted by sampling from the posterior distribution q_ω which is defined as a Normal distribution that is parameterized by (μ, σ). On the other hand, the decoder p_θ projects the $z_{\cdot t}$-codes to the deformation field ϕ_t while being conditioned on the moving image I_0. The output of the decoder are the reference image I_0 warped with the ϕ_t deformation fields. The decoder represents data likelihood of the latent variable model. Using a prior distribution $p(z)$ over latent variables z, we define the following generative process:

$$p_\theta(I_{1:T}|I_0) = \int_z p_\theta(I_{1:T}|z, I_0)p(z)\ dz, \tag{1}$$

which is visualized in Fig. 1a. In this work, encoder q_ω and decoder p_θ are approximated using neural networks where ω and θ represent the encoder and decoder networks' weights which are optimized using amortized Variational Inference [36]. The data likelihood $p_\theta(I_{1:T}|z, I_0)$ can be seen as the fidelity of the reconstruction of the fixed images $I_{1:T}$ by warping the moving image I_0 with appropriate deformations $\phi_{1:T}$. An overview of the motion model can be seen in Fig. 2.

1) Gaussian process prior: The prior follows a zero-centered multivariate Gaussian distribution: $p(z) \sim \mathcal{N}(0, \Sigma)$ where the covariance matrix Σ is a diagonal block matrix of dimensions $DT \times DT$:

$$\Sigma = \text{Diag}_{d=1}^{D}(K_l). \tag{2}$$

Each diagonal element of Σ represents the temporal covariance matrix $K_l \in \mathbb{R}^{T\times T}$ of a Gaussian time-continuous stochastic process whose kernels can be chosen by the user. A typical choice in Gaussian processes is the squared exponential kernel

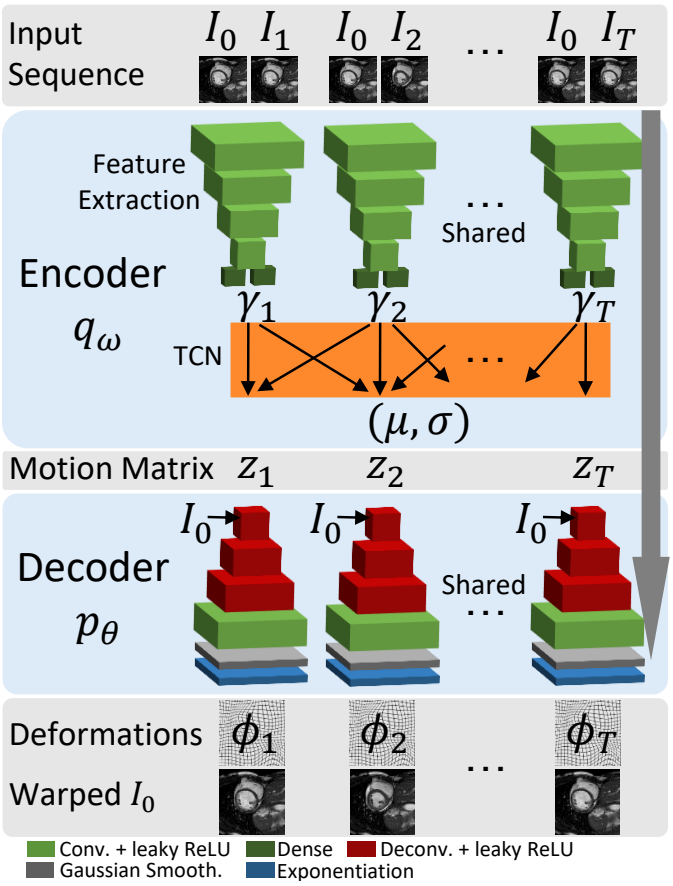


Fig. 2. Overview of the motion model including encoder and decoder neural networks. From sequential image pairs, temporally independent feature vectors γ_t are extracted which are fed to a temporally convolutional network (TCN) to obtain the probabilistic motion matrix z. This compact representation is decoded to a sequence of diffeomorphic deformation fields ϕ_t.

$K_l^{\text{RBF}}(\tau, \tau') = \sigma_K^2 \exp\left(-|\tau - \tau'|^2/2l^2\right)$ with length-scale l and variance σ_K^2. The length-scale practically describes, how close two points τ and τ' have to be to influence each other significantly. Therefore, it represents a number or range of time steps. Due to the fact that we want to model data that changes slowly and consistently over time we consider the Cauchy kernel [39], [40] that is heavy-tailed and has a long-range influence:

$$K_l^{\text{Cauchy}}(\tau, \tau') = \sigma_K^2\left(1 - \frac{(\tau - \tau')^2}{l^2}\right)^{-1}, \tag{3}$$

with pre-defined σ_K. This covariance matrix Σ allows temporally correlated latent variables while still assuming highest possible independence between the D latent dimensions. In other words, we extended the standard VAE latent space which only consists of the independence assumption between latent variables with a regularized temporal dimension. Latent variables are related over time according to the chosen kernel function K_l while being independent of each other. An example of a covariance matrix can be seen in Fig. 1b.

2) Posterior and Likelihood Distributions: Similar to standard VAEs, the posterior q_ω follows a multivariate Gaus-

sian distribution $q_\omega(z|I_{0:T}) \sim \mathcal{N}(\mu, \Sigma^*(\sigma))$ with data-driven predictions of mean vector $\mu \in \mathbb{R}^{DT}$ and variance vector $\sigma \in \mathbb{R}^D$. The full covariance matrix $\Sigma^*(\sigma)$ is defined as a D-dimensional block diagonal matrix of the following form where each block is scaled using the values of the variance vector σ:

$$\Sigma^*(\sigma) = \begin{bmatrix} \sigma_1 K_l & 0 & \cdots & 0 \\ 0 & \sigma_2 K_l & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_D K_l \end{bmatrix}. \tag{4}$$

Mean and variance vectors (μ, σ) are the output of the encoder neural network. The kernel K_l is kept the same as in the prior distribution and does not contain predicted parameters to guarantee a user-chosen temporal regularity.

Also, the likelihood p_θ is assumed to follow a multivariate Gaussian distribution $p_\theta(I_{1:T}|z, I_0) \sim \mathcal{N}(I_0 \circ \phi_{1:T}(\theta), \sigma_L * I_{DT})$ where the mean is represented by the warped moving image I_0 that is deformed by the diffeomorphisms $\phi_{1:T}$ (the decoder output) and $\circ$ denotes the image warping operation. The covariance is represented by the identity matrix I_{DT} of size DT times (denoted by $*$) the scalar σ_L which can depict for example the variance of intensity residuals of well registered images.

3) Learning the Motion Model via Variational Inference: In order to optimize the parameterized motion model over ω and θ, the evidence lower bound (ELBO) of the log-marginalized likelihood $p_\theta(I_{1:T}|I_0)$ that is conditioned on the moving image I_0, must be maximized (see [15], [34], [36] for details):

$$\log p_\theta(I_{1:T}|I_0) - \mathrm{KL}\big[q_\omega(z|I_{0:T})\|p(z|I_{0:T})\big] = \mathbb{E}_{z\in q_\omega(\cdot|I_{0:T})}\big[\log p_\theta(I_{1:T}|z, I_0)\big] - \mathrm{KL}\big[q_\omega(z|I_{0:T})\|p(z)\big], \tag{5}$$

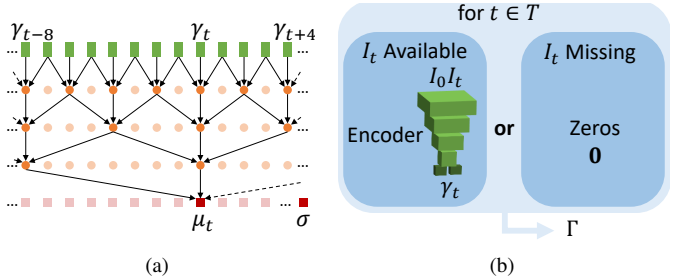
with KL denoting the Kullback-Leibler Divergence (KL). The first term in 5 enforces that the moving image I_0 is well registered to the fixed images $I_{1:T}$ by maximizing the log likelihood. The second term structures the latent motion encoding by enforcing the posterior distribution $q_\omega(z|I_{0:T})$ to be close to the prior distribution $p(z)$. Following the definition of the KL divergence between 2 multivariate Gaussian distributions, we obtain the closed-from solution (see Appendix A):

$$\mathrm{KL}\big[q_\omega(z|I_{0:T})\|p(z)\big] = \frac{1}{2}\sum_{i=1}^{D} \sigma_i^2 T + \bar{\mu}_i^\top K^{-1} \bar{\mu}_i - \log(\sigma_i^2) - T, \tag{6}$$

with $\bar{\mu}_i$ being the i-th segment of length T in μ.

Recall that the log likelihood $p_\theta(I_{1:T}|z, I_0)$ is also Gaussian. Thus, $\log p_\theta(I_{1:T}|z, I_0) = -\frac{1}{2}\sum_{t=1}^{T} \|I_t - I_0 \circ \phi_t\|^2/\sigma_L + C$ with a constant C which is equivalent to adopting a sum-of-squared differences (SSD) criterion, commonly used as similarity metric in image registration (for example in [41]).

During training of the model, parameters ω and θ are updated via stochastic gradient descent and back-propagation. In order to back-propagate through the sampling operation, the reparameterization trick is used [36]. For full-covariance Gaussian distributions, the covariance matrix must be positive-definite as we use the Cholesky decomposition for the reparameterization (cf. [42]). The details on how to efficiently compute the Cholesky decomposition of the covariance matrix Σ^* in 4 can be found in Appendix B. Note that the covariance matrix grows quadratically with the sequence length. This could potentially lead to high computational expenses for longer image sequences. However, due to the diagonal block structure and the possibility of precomputing the Cholesky decomposition (Appendix B), computations are very efficient. Especially for common image sequence length in medical imaging. For sequences of 30-50 images, the covariance computations are computationally not relevant in comparison to the rest of the neural networks.



(a) (b)

Fig. 3. (a) The temporal convolutional network (TCN) allows for temporal regularization of the independently extracted features γ_t per time step t, for retrieving mean vector μ and variance vector σ of the posterior distribution p_θ. (b) Sequences with missing time steps (motion interpolation or simulation) are encoded by a full feature matrix Γ by setting the columns of missing time steps to zero. The TCN handles these missing columns and still predicts a full temporal motion sequence of T time steps.

4) Neural Network Architecture: A graphical summary of the temporal encoder-decoder neural network architecture is shown in Fig. 2. The encoder takes the image pairs (I_0, I_t) as input and outputs the motion matrix z. It consists of a feature extraction part and a temporal regularizer (TCN). The feature extraction part consists of three down-sampling and one size-maintaining convolutional layers and one fully-connected layer of size $2D$ for low-dimensional mean and variance predictions of the posterior [34]. As non-linearities, leaky rectifier functions [43] have been used in the convolutional layers while the fully-connected layer is linear. These layers are temporally independent and share weights across all image pairs of a sequence. The output of the feature extraction networks is the feature matrix Γ with feature vectors γ_t per time step t of size $\mathbb{R}^{2D}$. These feature vectors are temporally regularized by merging them across different time steps using a temporal convolutional network (TCN). This leads to temporally consistent mean and variance vectors (μ, σ) that define the posterior distribution $q_\omega(z|I_{0:T}) \sim \mathcal{N}(\mu, \Sigma^*(\sigma))$. The size of $2D$ is chosen for γ_t such that each σ value can be influenced by features from the whole sequence. Note, that samples from the posterior distribution are vectors of size DT which are reshaped to retrieve the motion matrix z with $z_{.t}$-columns.

Following the recommended architecture, the TCN consists of 1D convolutional layers with increasing dilation and skip connections allowing to learn temporal dependencies of the latent variables γ_t that were time-independent before [38]. We use zero-padding and non-causal convolutional layers to

also take future time steps into account. In particular, four consecutive blocks of convolutions are used each consisting of a 1D convolution with 3x3 filters with rectifier non-linearities, a spatial 1D dropout layer and followed by a 1x1 linear convolution layer. In addition each block's output is added to their input to establish skip connections. An additional 1x1 linear convolutional layer is used as the first TCN layer and the output of the TCN is the sum of the outputs of all blocks. Size-maintaining zero-padding is used and the number of filters is $2D$ for all TCN layers to keep input and output matrices of same dimensions. The output tensor of size $\mathbb{R}^{DT+D}$ is split into μ and σ vectors where exponentiation is applied on the σ-vector to guarantee non-negative values close to 1. Our TCN is shown in Fig. 3a. TCNs can handle sequences of varying time lengths and are advantageous compared to recurrent neural networks (RNN) due to a flexible receptive field and more stable gradient computations [38]. Another reason why the authors chose a TCN over RNNs is that RNNs are especially suitable to learn long-distance temporal relationships such as in natural language processing while the focus of this work is on rather short time sequences with higher local dependencies. One could use a cyclic padding instead of zero-padding for cyclic sequences, for example by linking the end of a sequence to its beginning. However, in the case of cardiac cine-MRI, 5-10% of the cardiac cycle are often omitted [1] such that we chose to not assume cyclic sequences explicitly.

For each time step, the decoder takes as input the $z_{\cdot t}$ vector which are sampled from the posterior distribution by using the reparameterizaiton trick and outputs the diffeomorphisms $\phi_{1:T}$ and the accordingly warped moving image. A fully-connected layer, three up-sampling deconvolutional and two size-maintaining convolutional layers are used in the decoder which are shared across all time steps. The $z_{\cdot t}$ is first extended and reshaped in order to fit the input size of the first deconvolutional layer. It is desired that the latent representation z encodes deformation information on a semantic level, independent of the given subject. That is why the decoder is further conditioned on the moving image I_0 by concatenating down-sampled versions of I_0 with the outputs of the deconvolutional layers at different scales. The image I_0 is hereby down-sampled by tri-linear interpolation with factors 2, 4 and 8 while the original sized I_0 is concatenated after the third deconvolutional layer. By providing subject-specific appearance information in form of the moving image, the motion model is driven to encode subject-independent deformation information in the limited dimensionality of z [15]. Leaky rectifier functions [43] are used in the deconvolutional layer while a *tanh* activation is applied after the last convolutional layer for stability reasons during training.

In addition, a diffusion-like regularization in spatial and temporal dimensions is applied by Gaussian smoothing kernels. This regularization follows the derivations of [15] and is omitted in Fig. 1a for reasons of clarity. We utilize a Gaussian smoothing layer with standard deviations of σ_G and σ_T in temporal and spatial domains respectively. To ensure diffeomorphic deformations, an exponentiation layer [15] that relies on the *scaling-squaring* algorithm [18] for the stationary velocity field parameterization of diffeomorphisms is applied. The differentiable linear warping functionality is realized using a spatial transformer network layer [44]. The full details of architecture and training can be found in the implementation details section III-B. Additionally, a table summarizing all the layers is presented in Appendix C.

B. Missing Data and Temporal Dropout

The temporal latent dimensionality T (the size of the covariance matrix Σ^*) is kept identical across datasets with different time lengths T^*. However, the model can handle arbitrary lengths of sequences up to this maximum length T which needs to be set before training starts (e.g. the maximum expected sequence length). In case of shorter sequences, the features γ_τ of all available image pairs (I_0, I_τ) with $\tau \in T^*$ are extracted and evenly distributed along T forming the matrix $\Gamma \in \mathbb{R}^{2D \times T}$. The remaining missing time steps are filled with a constant (typically zero). As typical in handling of incomplete data in neural networks, in the course of training, this constant will be associated to as missing time steps as these values are not beneficial for optimizing the loss function. On the decoder side, the log-likelihood loss (first part of 5) is evaluated on all available time steps of the original sequence. If a sequence is longer than T, evenly distributed frames would be dropped to reach a length of T. However, this should not happen normally as we assume to put T at least as the maximum experienced length in the data.

In addition, during training, further time steps (i.e. γ_τ) are dropped from Γ using temporal dropout (TD) in order to force the motion model to interpolate motion between available frames. To encourage the TCN to make use of its temporal connections and search for dependencies across time, our TC drops some of the γ_τ while still trying to recover the deformations ϕ_τ of all available image pairs (I_0, I_τ). More precisely, in TD, instead of extracting features from an image pair (I_0, I_τ), a vector of zeros is chosen as γ_τ while still keeping the loss function on the decoder part for these time steps. A binary Bernoulli random variable r_τ is used to randomly choose at each original time step τ if the zero vector is used instead of the extracted features given (I_0, I_τ). All independent Bernoulli random variables $r \in \mathbb{R}^{T^*}$ have the success probability δ. The latent feature representation γ_t^{TD} using TD can thus be defined as:

$$\gamma_\tau^{TD} = r_\tau * \mathbf{0} + (1 - r_\tau) * \gamma_\tau. \tag{7}$$

Note, TD is used only during training as a sort of self-supervision to encourage generalizability and consistent motion simulation and interpolation of missing data. When encountering missing data at test time, one just needs to place the available encoded frame pairs at the desired temporal positions of Γ in order to predict the full motion consisting of T time steps (cf. Fig. 3b). A full motion simulation can be generated by setting all elements of Γ to zero. In this case, a sequence of deformations that are plausible with respect to the training data will be predicted given only the original image I_0.

Optional Random Sub-Sequence Training: Since our motion model takes sequences of images as input and outputs a sequence of deformation fields, it comes naturally with high

computational costs. This can lead to a model that may not be trainable on standard GPUs. Due to this limitation, we propose to train our model optionally with random sub-sequences. This can be done by dropping the encoder and decoder for some time steps while keeping the full temporal dimensionality in the latent space (the motion matrix and TCN). Let $\mathcal{T}$ be the maximum number of frames with which our model can be trained on a given GPU. In each training iteration, a random combination of $\mathcal{T}$ frames is selected from a training subject with T^* frames whenever $T^* > \mathcal{T}$. As in the case of missing data, the covariance matrix and TCN is kept with the original size containing T time steps. The selected frame pairs are encoded and placed at their relative temporal position in Γ while filling the remaining time steps with zeros. In contrast to the TD or motion interpolation procedure, only the selected $\mathcal{T}$ time steps are reconstructed in the decoder to limit the requirements of GPU memory. In case of shorter training sequences with $T^* \leq \mathcal{T}$, the full sequence is used. By sampling different sub-sequences in each training epoch, the network will eventually see all parts of a sequence during the training stage.

III. Experiments

In this paper, we evaluate the proposed motion model on cardiac cine-MRI. Besides accurate temporal tracking and registration, we show the model's capabilities for motion simulation, interpolation and transport. The improved temporal latent space using the GP prior is demonstrated. Extensive results are presented for 2D+t sequences with more limited quantitative evaluations on 3D+t sequences due to their heavy computational requirements. In all experiments, the end-diastolic (ED) frame was used as the moving image I_0.

A. Data sets

Two data sets forming 334 cardiac cine-MRI in total were used. First, 184 multi-centric short-axis sequences came from the EU FP7-funded project MD-Paedigree (Grant Agreement 600932), with congenital heart disease and healthy or pathological images from adults. In addition, 150 sequences originated from the Automatic Cardiac Diagnosis Challenge 2017 (ACDC [1]). The images were acquired in breath hold using 1R-R or 2R-R intervals mixing retrospective or prospective gating. The original sequence lengths varied from 13 to 35 frames. The 100 *training* cases from ACDC that contain ED-ES segmentation information were used for testing while all other sequences were used for training. Slices were resampled with a spacing of 1.5×1.5 mm and cropped to a size of 128×128 pixels. In case of 3D+t sequences, 18 slices were used by adding zero slices at the top and bottom in case of fewer original slices.

B. Implementation Details

The feature extractor consisted of 4 convolutional layers with (2,2,2,1)-strides and (16,32,32,4)-feature maps and a fully-connected layer of size $2D$, outputting γ_t. The decoder p_θ consisted of 3 deconvolutional and 1 convolutional layer with (32,32,32,16)-feature maps. All (de-)convolutional layers in encoder and decoder used a kernel size of 3. The TCN consisted of four 3x3 1-D convolutional blocks with (1,2,4,8)-dilations. The spatio-temporal Gaussian layer used the spatial $\sigma_G = 3$mm and temporal standard deviation $\sigma_T = 1.5$, the exponentiation layer used 6 *scaling-squaring* iterations computed using the formula in [18]. The latent dimensionality was set to $D = 32$ (as in [15]). We set the sequence length T to 35, the maximum sequence length found in the training data, resulting in a motion matrix z with $D \cdot T = 1088$ elements. The frames of shorter sequences were evenly distributed over T time steps and the gaps were marked as missing data as described in section II-B. The number of trainable parameters (ω, θ) in the network summed up to ∼210k in 2D+t and ∼456k in 3D+t respectively. L2 weight decay of $1 \cdot 10^{-4}$ was applied on all layers The Cauchy-kernel parameters were chosen as proposed in [40] with $l = 7$ and $\sigma_K = 1.005$. The variance of the data likelihood was set as the variance of intensity residuals of a few well-registered image sequences with $\sigma_L = 0.0045$ in 2D+t and 0.00021 in 3D+t respectively. For further details of the neural network architecture, the reader is referred to Appendix C.

For training, we used a first-order gradient-based method for stochastic optimization (Adam [37]) with a batch size of one and fixed learning rate of 0.00015. The TD probability δ was 0.5. Random sub-sequence training was only applied for 3D+t with $\mathcal{T} = 18$. Online data augmentation containing randomly shifted, rotated, scaled and mirrored images has been applied to increase generalizability of the model. The model was implemented using Keras [45] and Tensorflow [46]. The training time was ∼15h in 2D+t and 7 days for 3D+t sequences on a NVIDIA GTX TITAN X GPU.

C. Registration and Motion Prediction

We compare our model in terms of registration accuracy and spatio-temporal deformation regularity with 3 state-of-the-art diffeomorphic methods: SyN [10], the learning-based probabilistic pairwise registration (LPR [15]) and the temporal B-spline algorithm in elastix (4D-Elastix [22]). We also compare with the previous version of our method without Gaussian process prior (No-GP [35]). SyN and 4D-Elastix have been manually tuned on a few training images following the recommendations in the original papers. The LPR algorithm has been trained on a 2D single scale version using all image pairs of a sequence instead of only the end-diastolic/end-systolic (ED, ES) pairs. We measured registration accuracy using the root mean square error (RMSE) of intensities and segmentation-based DICE scores and 95%-tile Hausdorff distances (HD, in mm) on the five anatomical structures available in ACDC: left ventricle myocardium (LV-Myo), epicardium (LV), left ventricle bloodpool (LV-BP), right ventricle (RV) and LV+RV. In terms of registration regularity, we report spatial (Spatial Grad.) and temporal gradients (Temp. Grad.) of the deformation fields ϕ_t with $t \in [1, T]$.

The reported results in Table I were measured on all 2D test sequences containing at least one mask (resulting in 677 sequences from 100 test subjects). DICE scores and Hausdorff

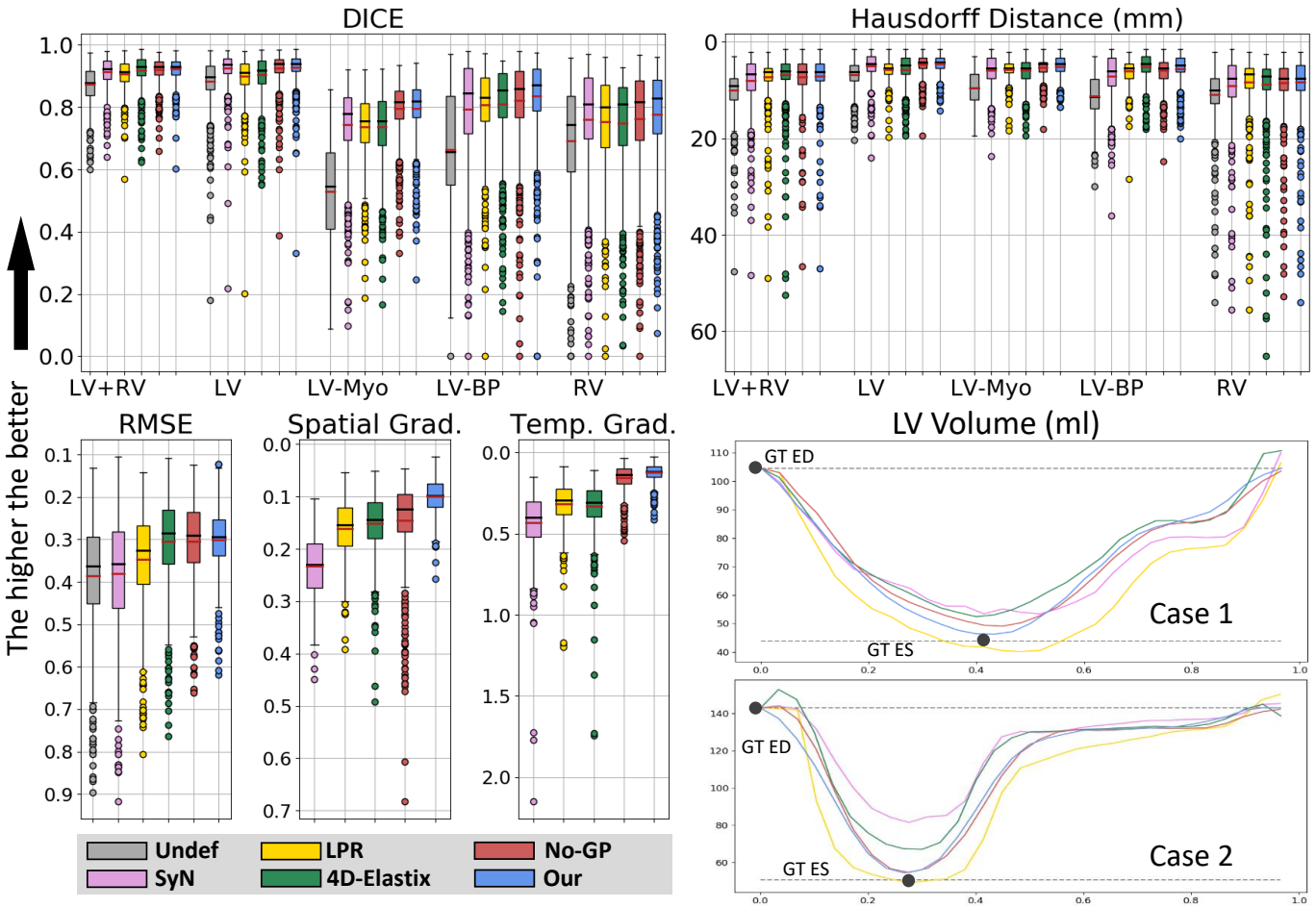


Fig. 4. Tracking results showing RMSE, spatial and temporal gradients of the displacement fields, DICE scores and Hausdorff distances for all 2D+t test sequences. The LV volume curves extracted from the warped ED blood pool masks for 2 random test cases in ml, show the temporal smoothness and the distance to the ground-truth ED and ES volumes (marked with black points). The proposed algorithm (Our) shows slightly higher registration accuracy and temporally smoother deformations than the state-of-the-art algorithms: SyN [10], LPR [15], 4D-Elastix [22] and the previous version of our method without GP prior (No-GP [35]).

TABLE I
REGISTRATION PERFORMANCE WITH MEAN AND STANDARD DEVIATION SCORES OF DICE (IN %), HAUSDORFF DISTANCE (HD IN MM), SPATIAL AND TEMPORAL GRADIENTS OF THE DEFORMATION FIELDS ($\times 10^{-2}$) COMPARING OUR METHOD WITH THE UNDEFORMED CASE (UND), SYN, LEARNING-BASED PAIRWISE REGISTRATION (LPR), 4D-ELASTIX (4D-E) AND OUR PREVIOUS VERSION WITHOUT GP PRIOR (NO-GP) IN 2D+T.

Method	DICE	HD	Spat. Grad.	Temp. Grad.
Und	72.8 ±14	9.70 ±4.20	–	–
SyN	82.7 ±12	7.02 ±4.34	0.23 ±0.06	0.43 ±0.19
LPR	82.1 ±10	6.60 ±3.07	0.16 ±0.06	0.32 ±0.13
4D-E	83.7 ±11	6.27 ±3.91	0.15 ±0.06	0.33 ±0.15
No-GP	84.6 ±10	6.24 ±3.30	0.14 ±0.08	0.15 ±0.08
Our	85.2 ±09	6.11 ±3.28	0.10 ±0.03	0.12 ±0.05

distances are only reported for the frames with available ground-truth segmentation (ES images). Detailed box plots of the results together with LV volume curves are shown in Fig. 4. The LV volumes (in ml) were extracted by warping the ED mask according to the extracted deformation fields and computing the blood pool volume for all slices of one subject over time. The results indicate that our model achieves the same (RMSE) or slightly better (DICE and HD) registration accuracy compared to the reference methods while improving spatial and temporal regularity as shown by the deformation field gradients and the volume curves.

TABLE II
3D+T REGISTRATION PERFORMANCE WITH MEAN AND STANDARD DEVIATION SCORES OF RSME, DICE, HAUSDORFF DISTANCE (HD), SPATIAL AND TEMPORAL GRADIENTS OF THE DEFORMATION FIELDS COMPARING THE UNDEFORMED CASE (UND), 4D-ELASTIX (4D-E) AND THE PROPOSED METHOD.

	DICE	HD	Spat. G.	Temp. G.
Und	70.1 ±12	7.7 ±2.7	–	–
4D-E	79.2 ±10	5.1 ±2.1	0.15 ±0.06	0.62 ±0.32
Our	79.5 ±09	5.4 ±2.1	0.07 ±0.02	0.09 ±0.03

In Table II, we show the results on the 100 test sequences for our 3D+t model. In comparison to 4D-Elastix, our 3D+t model shows a similar registration accuracy but a significantly im-

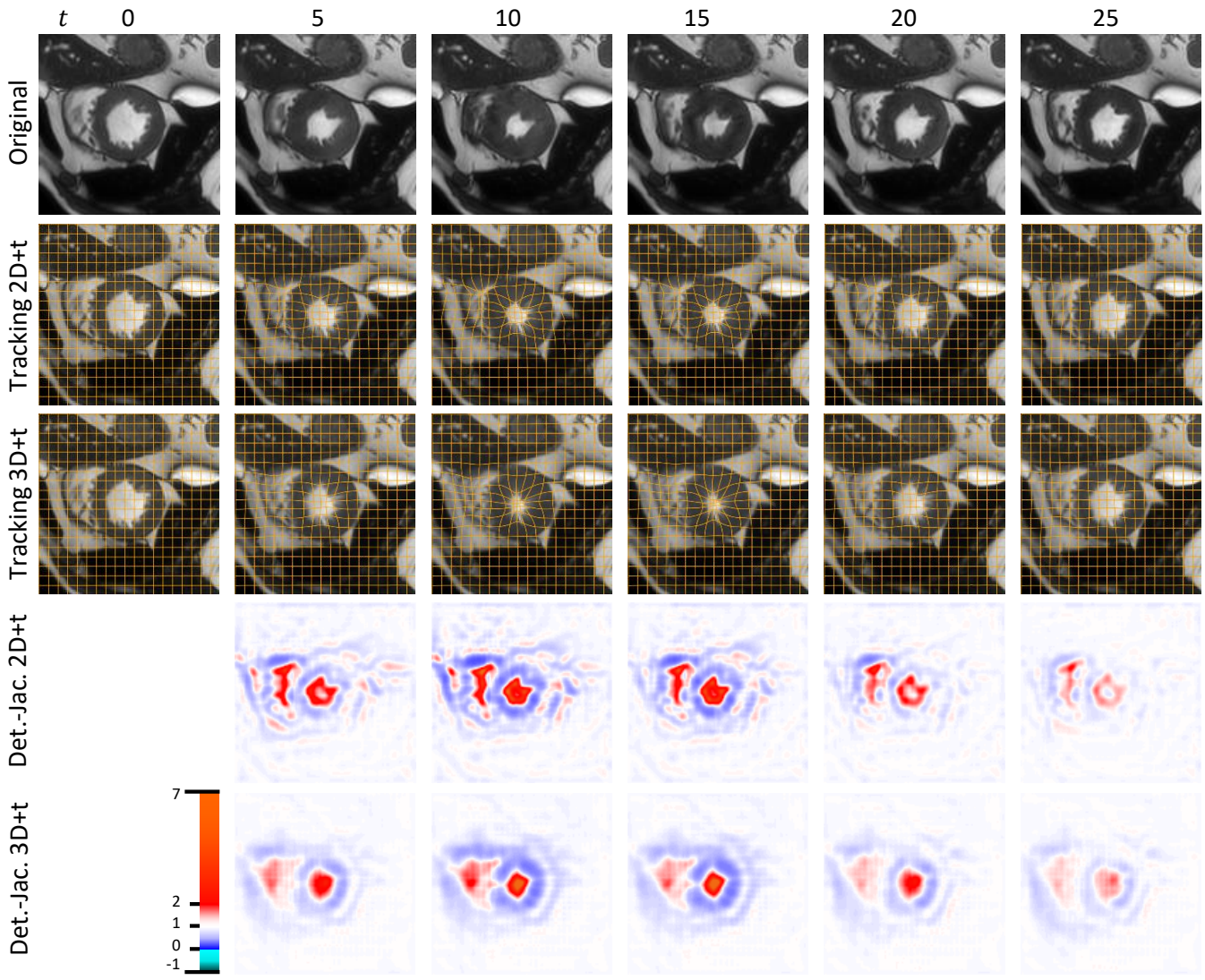


Fig. 5. Showing 2D+t and 3D+t tracking results of the warped moving image I_0 with grid overlay and the Jacobian determinant (Det.-Jac.) for the mid-ventrical slice of a test sequence. In 3D+t, smoother Jacobian determinants were obtained.

proved spatial and temporal regularity. In addition, our model has a lower RMSE with 0.16 ± 0.05 compared to 4D-Ealstix with 0.18 ± 0.07. In Fig. 5, the warped moving image I_0 and the Jacobian determinant are visualized for one test sequence in 2D+t and 3D+t. One can see, the Jacobian determinants are smoother in 3D+t compared to 2D+t sequences.

The new Gaussian process prior leads to smoother deformations compared to the previous time-independent prior (No-GP version) while using the same deformation field regularizer. This can be also seen in Fig. 6 where the first 5 latent dimensions, the sequences $z_{d\cdot}$ with $d \in [0, 4]$, are visualized for one test case. Furthermore, we investigated the insensitivity of our motion model with respect to initial alignments of the test sequences with the motion model. To this end, we rotated all test sequences by 0, 90, 180 and 270 degrees and compared the performance in Appendix D. We found no statistical significant differences between the results of the 4 test runs demonstrating the orientation independence of the motion model.

D. Motion Simulation, Interpolation and Transport

To evaluate the performance on motion interpolation and simulation, we challenged our model to predict the motion for all time steps from a limited number of input frames. Thus, the goal was to predict motion patterns that are as close as possible to the observed motion of the full sequence (i.e. all registered frames obtained in the all frame model of the previous section III-C). Just as in temporal dropout during training, all the missing frames were represented as zero columns γ_t in the feature matrix Γ as shown in Fig. 3b. We compared the motion predictions from various input frame subsets that are provided to the model. First, we provided every 2nd or every 5th frame for motion interpolation. Then, we provided the first 5 frames or only the 10th frame (0th + 10th) to see if the model is able to complete typical cardiac motion patterns. Finally, we tested the full motion simulation by letting the model find a

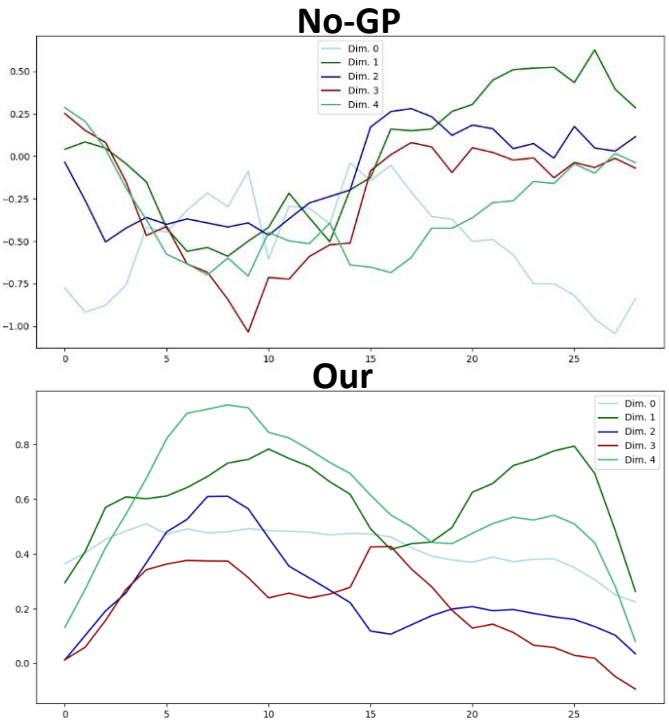


Fig. 6. First 5 latent dimensions of the same test sequence shows a temporally smoother motion matrix z (sampled from posterior given μ and σ) for the proposed model trained with the Gaussian process prior compared to the No-GP version.

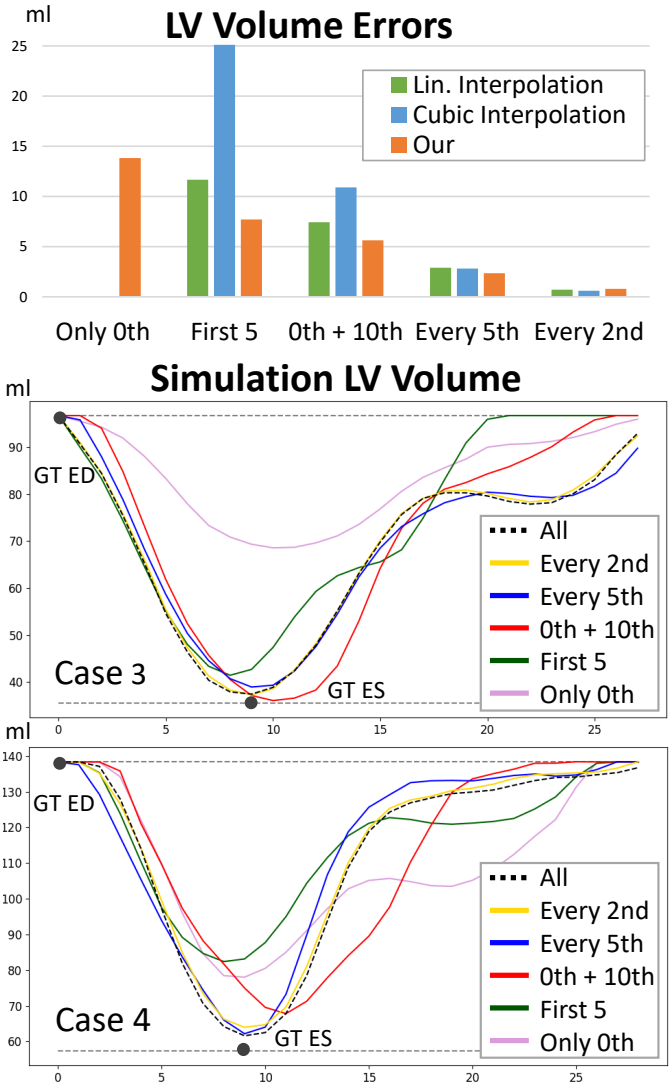


Fig. 7. Predicted simulated and interpolated motion from a limited number of frames. Provided frames are decreasing from all frames to only the 0th frame (full motion simulation). The volume errors with respect to the all frame prediction are compared with linear and cubic interpolation of the deformation fields. Two random test subjects are shown in the bottom.

motion sequence given only the moving image I_0 (only 0th) and setting feature matrix Γ to zero everywhere. We compared the simulated motion, with linear and cubic interpolation of the deformation fields (which are taken from the all frame model at the selected time steps). In the top of Fig. 7, average LV volume errors (RMSE) with respect to the all frame model were computed for all 677 test sequences in comparison to linear and cubic interpolation. In the bottom of Fig. 7, one can see the results of our model for the different interpolation cases in terms of LV volume curves for two example sequences.

For the cases of providing every 2nd and every 5th frame, our model interpolated the motion similarly well as linear or cubic interpolation, while providing better results in the cases of providing the 0th+10th and first 5 frames signaling an improved learned cardiac motion model. The full simulation (only 0th) did not result in well fitted volume curves, which is expected as the model has to simulate the full motion sequence from just the ED frame. However, it is observable that the model learned realistic cardiac specific motion patterns as the volume curves for example show the plateau phase before atrial systole which can be also seen in the completed motion for the cases where we provide the first 5 and 0th+10th frames. For the full simulation, our model often slightly underestimated the motion (cf. case 3 in Fig. 7) which can be related to the pathology distribution in the training dataset which contained many cases with reduced cardiac motion.

Furthermore, we demonstrate the model's capacity of motion transport in a qualitative way. Our model allows to transport motion patterns from one subject to another by taking the motion matrix z of one case and applying it on the moving image of another image sequence (ED frame). In this way, for example a pathological motion can be simulated in a healthy subject or vice versa. In Fig. 8, we present 2 subjects from the ACDC dataset, from which one is classified as healthy and the other as a dilated myopathy case (DCM). We extracted the motion matrices for both and applied them on the ED frame of the other case, such that we simulated a DCM typical motion in the healthy case while *curing* the pathological case. This can be seen for example from the LV contraction strengths in the Jacobian determinants or the related ejection fraction (EF). Note, that this form of parallel transport does not require any additional inter-subject registration.

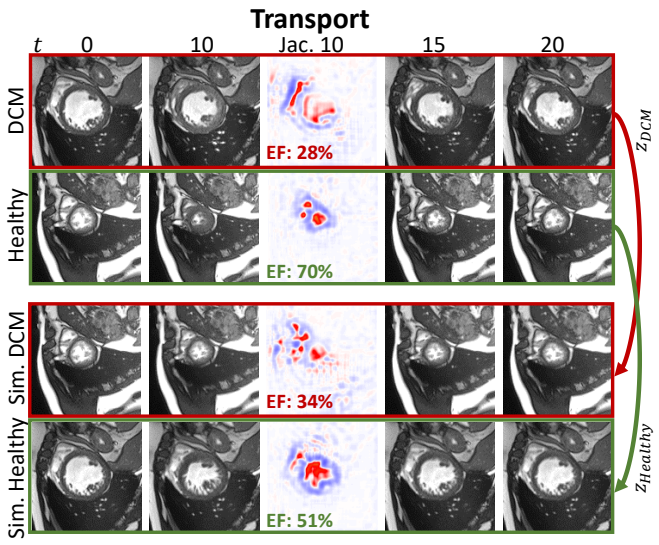


Fig. 8. Transporting the motion matrix z from one subject and combining it with the end-diastolic frame of another subject allows for simulating a disease (dilated myopathy, DCM, red motion) in a healthy subject and vice versa (green motion). Ejection fraction (EF) of the simulated cases (Sim. DCM and Sim. Healthy) are more similar to the original transported motion.

IV. Discussion

Our approach has shown state-of-the art registration accuracy and improved deformation regularity temporally and spatially in comparison to 3 state-of-the-art algorithms indicating that the low-dimensional motion encoding helps to regularize the registration problem of cardiac image sequences. We have shown that the novel Gaussian process prior leads to a higher temporal consistency compared to the time-independent prior [35] both, in latent and deformation space. A temporally smoother latent space is desirable as it brings more structure and interpretability and is consistent with the temporally smooth motion we experience in deformation space. We have demonstrated motion simulation and interpolation from a very limited number of frames indicating that data acquisition could be speed up as fewer frames are required in order to retrieve an accurate motion. In case of full simulations, our model showed a slightly reduced cardiac motion compared to healthy subjects. The authors believe this is due to a bias introduced from the disease distribution in the training data. To not end up with such a mean motion that merges several pathological motion patterns, one could think of generating disease-specific models. This could be achieved by training different motion models with training sets separated by diseases. We assumed intensities to be constant within an image sequence by using a Gaussian log-likelihood distribution (SSD criterion). Contrast variations can be handled for example by deploying a local cross-correlation distribution as in [15]. However, we found that in the given use case the SSD criterion worked slightly better. Thus, we chose the Gaussian likelihood distribution in this work. As another extension to our previous work, we have shown first results on 3D+t sequences which showed smoother Jacobian determinants than the 2D+t version which can be explained by out-of-plane deformations and the fact that we kept the latent dimensionality D the same for 2D and 3D versions of our algorithm. As the full deformation fields in 3D have more parameters than in 2D the results reconstructed from latent parameters are more smoothed. A lower amount of smoothness and more deformation details could be reached by increasing the latent dimensionality D. However, a limitation is the high computational costs for 3D+t sequences with long training times even for relatively low-dimensional images.

In future work, we aim to reduce this complexity and show the approach's potential generalizability to other applications such as for example respiratory motion estimation in dynamic images of the lung. A natural limitation of the proposed model is its low-dimensional latent representation that could become a bottleneck when facing more difficult deformation patterns in other use cases. Also, if the experienced variance in motion in the training data becomes larger, the model first requires more training data but may also encounter difficulties in finding a reasonable latent representation that is able to capture all the variations in the training data.

V. Conclusion

We presented a probabilistic motion model learned from images that can be useful for spatio-temporal registration, temporal super-resolution, data augmentation, shorter acquisition times and motion analysis for cardiac cine-MRI. Based on a novel Gaussian process prior conditional variational autoencoder, the model captures intrinsic motion patterns encoded in a low-dimensional probabilistic space – the motion matrix. We have shown that such a space allows for accurate diffeomorphic tracking, temporal interpolation, motion simulation and motion transport. The authors believe the presented application-specific motion model that does not rely on hand-crafted features such as bio-mechanical parameters could help in the understanding and analysis of moving organs such as the heart. The results indicate that it is possible to extract a small number of latent parameters in an unsupervised fashion to describe the cardiac motion without requiring much pre-processing of the image sequences. Furthermore, the authors believe the motion matrix as a compact representation of organ motion can be helpful as a quantitative new tool to guide the diagnosis, prognosis or therapy of diseases of dynamic organs such as the heart.

Disclaimer: The concepts and information presented in this paper are based on research results that are not commercially available.

Appendix

A. KL Divergence using the GP Prior

Given 2 multivariate Gaussian distributions with the same dimensionality, the KL divergence is defined in [47]. Suppose, we take our prior distribution $p(z)$ with zero-mean $\mathbf{0}$ and

covariance Σ of the form of 2 and our posterior distribution q_ω with mean μ and covariance Σ^* with dimensionality DT:

$$\mathrm{KL}[q_\omega(z|I_{0:T})\|p(z)] = \frac{1}{2}\left(\mathrm{tr}(\Sigma^{-1}\Sigma^*) + \mu^\top\Sigma^{-1}\mu - DT + \ln\left(\frac{\det\Sigma}{\det\Sigma^*}\right)\right). \quad (8)$$

The determinants of the block diagonal matrices Σ, Σ^* are $\det\Sigma = |K|^D$ and $\det\Sigma^* = |K|^D\prod_{i=1}^D\sigma_i^2$. Thus, the logarithm of the fraction of determinants in 8 becomes:

$$\ln\left(\frac{\det\Sigma}{\det\Sigma^*}\right) = \ln\left(\frac{1}{\prod_{i=1}^D\sigma_i^2}\right) = -\sum_{i=1}^D\ln\sigma_i^2 \quad (9)$$

When taking the sum over the D latent dimensions over the remaining terms, 8 simplifies to:

$$\mathrm{KL}[q_\omega(z|I_{0:T})\|p(z)] = \frac{1}{2}\sum_{i=1}^D\sigma_i^2 T + \bar{\mu}_i^\top K^{-1}\bar{\mu}_i - T - \ln(\sigma_i^2) \quad (10)$$

with $\bar{\mu}_i$ being the i-th segment of length T in μ. In the case of prior and posterior being identical, thus $\mu = \mathbf{0}$ and $\sigma = \mathbf{1}$ the quantity in 10 becomes 0.

B. Cholesky Decomposition of Σ^*

The Cholesky decomposition of a symmetric positive-definite matrix X equals the matrix product of a lower-diagonal L and its transposed: $X = LL^\top$. The entries of L can be computed by the Cholesky-Banachiewicz algorithm:

$$L_{j,j} = \sqrt{X_{j,j} - \sum_{k=1}^{j-1}L_{j,k}^2}$$
$$L_{i,j} = \frac{1}{L_{j,j}}\left(X_{i,j} - \sum_{k=1}^{j-1}L_{i,k}Lj,k\right) \qquad \text{for } i > j. \quad (11)$$

In case of the block diagonal matrix Σ^* the lower triangular matrix L^* equals a block diagonal matrix with lower triangular matrices that are resulting from the Cholesky decompositions of the diagonal block elements of Σ^*. Thus, in order to compute L^*, the Cholesky decompositions of the $i \in D$ diagonal elements $\sigma_i K$ must be computed. From Eq. 11 it follows that $c\cdot X = (\sqrt{c}\cdot L)(\sqrt{c}\cdot L^\top)$. Thus, $\sigma_i K = (\sqrt{\sigma_i}\cdot L_K)(\sqrt{\sigma_i}\cdot L_K^\top)$ and L^* is:

$$L^* = \mathrm{Diag}_{d=1}^D(\sqrt{\sigma_d}\cdot L_K). \quad (12)$$

Since the kernel matrix K is fixed in our framework, L_K can be pre-computed using 11 and reused keeping the computational efforts minimal even for a large covariance matrix Σ^*.

C. Network architecture

The neural network architecture is presented in Fig. V-B. The presented configuration was used for the 2D version of the proposed method. The architecture in 3D is identical just that the 3rd dimension with size 18 is added, thus using 3D convolutional respectively deconvolutional layers. This presents one possible architecture, other, for example deeper network are possible likewise.

Layer		Input Shape	Output Shape	Activation	# Filters /Nodes
Input			(B,T,128,128,2)	-	-
Apply for T time steps	Conv (3x3, stride 2)	(B, 128,128,2)	(B, 64,64,16)	LeakyReLU	16
	Conv (3x3, stride 2)	(B, 64,64,16)	(B, 32,32,32)	LeakyReLU	32
	Conv (3x3, stride 2)	(B,32,32,32)	(B, 16,16,32)	LeakyReLU	32
	Conv (3x3, stride 1)	(B,16,16,32)	(B, 16,16,4)	LeakyReLU	4
	Reshape	(B, 16,16,4)	(B,1024)	-	-
	Dense	(B, 1024)	(B, 64)	Linear	64
TCN		(B, T, 64)	(B, T, 64)	Linear	-
GP Sampling		(B,T,64)	(B,T,32) (motion matrix)	-	-
Apply for T time steps	Dense	(B,32)	(B, 1024)	LeakyReLU	1024
	Reshape	(B, 1024)	(B,16,16,4)	-	-
	Concatenate (with I_0)	(B,16,16,4)(B,16,16,1)	(B,16,16,5)	-	-
	Deconv (3x3, stride 2)	(B,16,16,5)	(B,32,32,32)	LeakyReLU	32
	Concatenate (with I_0)	(B,32,32,32)(B,32,32,1)	(B,32,32,33)	-	-
	Deconv (3x3, stride 2)	(B,32,32,33)	(B,64,64,32)	LeakyReLU	32
	Concatenate (with I_0)	(B,64,64,32)(B,64,64,1)	(B,64,64,33)	-	-
	Deconv (3x3, stride 2)	(B,64,64,33)	(B,128,128,32)	LeakyReLU	32
	Concatenate (with I_0)	(B,128,128,32)(B,128,128,1)	(B,128,128,33)	-	-
	Conv (3x3, stride 1)	(B,128,128,33)	(B, 128,128,16)	LeakyReLU	16
	Conv (3x3, stride 1)	(B, 128,128,16)	(B, 128,128,2)	tanh	3
	Gaussian Smoothing	(B, 128,128,2)	(B, 128,128,2) (velocities)	-	-
	Exponentiation	(B, 128,128,2)	(B, 128,128,2) (deformation)	-	-
	STN (with I_0)	(B, 128,128,2) (B, 128,128,1)	(B, 128,128,1) (warped I_0)	-	-
Output			(B, T, 128,128,1)	-	-

Fig. 9. Summary of the neural network architecture of the 2D version of the presented algorithm. Note that most layers are shared over time and are applied on all time instances T with shared weights. The batch size is denoted with B.

D. Alignment Sensitivity

TABLE III
REGISTRATION PERFORMANCE COMPARING THE TEST DATA SET ROTATED COUNTER-CLOCKWISE WITH 0, 90, 180 AND 270 DEGREES RESPECTIVELY (IN 2D-T). THE NETWORK HAS NOT BEEN RETRAINED. THE LAST ROW SHOWS AVERAGE AND STANDARD DEVIATIONS (SUMMARY) OF THE FOUR TEST RUNS.

	DICE	HD	Spat. G.	Temp. G.
0°	85.2 ±09	6.11 ±3.28	0.10 ±0.03	0.13 ±0.05
90°	84.9 ±09	6.24 ±3.27	0.10 ±0.03	0.13 ±0.05
180°	85.1 ±09	6.08 ±3.25	0.11 ±0.04	0.13 ±0.06
270°	85.0 ±09	6.10 ±3.19	0.11 ±0.04	0.13 ±0.06
summary	85.1 ±0.11	6.14 ±0.07	0.11 ±0.002	0.13 ±0.004

REFERENCES

[1] O. Bernard *et al.*, "Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: Is the problem solved?" *IEEE Transactions on Medical Imaging*, vol. 37, no. 11, pp. 2514–2525, 2018.

[2] J. Girija, G. K. Murthy, and P. C. Reddy, "4d medical image registration: A survey," in *2017 International Conference on Intelligent Sustainable Systems (ICISS)*. IEEE, 2017, pp. 539–547.

[3] M.-M. Rohé, M. Sermesant, and X. Pennec, "Low-dimensional representation of cardiac motion using barycentric subspaces: A new group-wise paradigm for estimation, analysis, and reconstruction," *Medical Image Analysis*, vol. 45, no. 1, pp. 1–12, 2018.

[4] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: A survey," *IEEE Transactions on Medical Imaging*, vol. 32, no. 7, pp. 1153–1190, 2013.

[5] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Diffeomorphic demons: Efficient non-parametric image registration," *NeuroImage*, vol. 45, no. 1, pp. S61–S72, 2009.

[6] J.-M. Peyrat, H. Delingette, M. Sermesant, C. Xu, and N. Ayache, "Registration of 4d cardiac ct sequences under trajectory constraints with multichannel diffeomorphic demons," *IEEE Transactions on Medical Imaging*, vol. 29, no. 7, pp. 1351–1368, 2010.

[7] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes, "Computing large deformation metric mappings via geodesic flows of diffeomorphisms," *International Journal of Computer Vision*, vol. 61, no. 2, pp. 139–157, 2005.

[8] M. Zhang and P. T. Fletcher, "Finite-dimensional lie algebras for fast diffeomorphic image registration," in *International Conference on Information Processing in Medical Imaging*. Springer, 2015, pp. 249–260.

[9] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Symmetric log-domain diffeomorphic registration: A demons-based approach," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2008, pp. 754–761.

[10] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee, "Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain," *Medical Image Analysis*, vol. 12, no. 1, pp. 26–41, 2008.

[11] M. Lorenzi, N. Ayache, G. B. Frisoni, and for the Alzheimer's Disease Neuroimaging Initiative (ADNI), "LCC-Demons: a robust and accurate symmetric diffeomorphic registration algorithm," *NeuroImage*, vol. 81, no. 1, pp. 470–483, 2013.

[12] X. Yang, R. Kwitt, M. Styner, and M. Niethammer, "Quicksilver: Fast predictive image registration–a deep learning approach," *NeuroImage*, vol. 158, no. 1, pp. 378–396, 2017.

[13] M.-M. Rohé, M. Datar, T. Heimann, M. Sermesant, and X. Pennec, "SVF-net: Learning deformable image registration using shape matching," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 266–274.

[14] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu, "Unsupervised learning for fast probabilistic diffeomorphic registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 729–738.

[15] J. Krebs, H. Delingette, B. Mailhé, N. Ayache, and T. Mansi, "Learning a probabilistic model for diffeomorphic registration," *IEEE Transactions on Medical Imaging*, vol. 38, no. 9, pp. 2165–2176, Sep 2019.

[16] B. D. de Vos, F. F. Berendsen, M. A. Viergever, H. Sokooti, M. Staring, and I. Išgum, "A deep learning framework for unsupervised affine and deformable image registration," *Medical Image Analysis*, vol. 52, no. 1, pp. 128–143, 2019.

[17] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu, "Unsupervised learning of probabilistic diffeomorphic registration for images and surfaces," *Medical Image Analysis*, vol. 57, no. 1, pp. 226–236, 2019.

[18] V. Arsigny, O. Commowick, X. Pennec, and N. Ayache, "A log-euclidean framework for statistics on diffeomorphisms," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2006, pp. 924–931.

[19] M. J. Ledesma-Carbayo *et al.*, "Spatio-temporal nonrigid registration for ultrasound cardiac motion estimation," *IEEE Transactions on Medical Imaging*, vol. 24, no. 9, pp. 1113–1126, 2005.

[20] J. Vandemeulebroucke, S. Rit, J. Kybic, P. Clarysse, and D. Sarrut, "Spatiotemporal motion estimation for respiratory-correlated imaging of the lungs," *Medical Physics*, vol. 38, no. 1, pp. 166–178, 2011.

[21] M. De Craene *et al.*, "Temporal diffeomorphic free-form deformation: Application to motion and strain estimation from 3d echocardiography," *Medical Image Analysis*, vol. 16, no. 2, pp. 427–450, 2012.

[22] C. Metz, S. Klein, M. Schaap, T. van Walsum, and W. J. Niessen, "Nonrigid registration of dynamic medical imaging data using nd+ t b-splines and a groupwise optimization approach," *Medical Image Analysis*, vol. 15, no. 2, pp. 238–249, 2011.

[23] C. Qin *et al.*, "Joint learning of motion estimation and segmentation for cardiac mr image sequences," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 472–480.

[24] W. Shi *et al.*, "Temporal sparse free-form deformations," *Medical Image Analysis*, vol. 17, no. 7, pp. 779–789, 2013.

[25] A. Giger *et al.*, "Respiratory motion modelling using cgans," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 81–88.

[26] A. Giger *et al.*, "Inter-fractional respiratory motion modelling from abdominal ultrasound: A feasibility study," in *International Workshop on PRedictive Intelligence In MEdicine*. Springer, 2019, pp. 11–22.

[27] J. Caballero *et al.*, "Real-time video super-resolution with spatio-temporal networks and motion compensation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4778–4787.

[28] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Video super-resolution with convolutional neural networks," *IEEE Transactions on Computational Imaging*, vol. 2, no. 2, pp. 109–122, 2016.

[29] M. Sermesant *et al.*, "Toward patient-specific myocardial models of the heart," *Heart failure clinics*, vol. 4, no. 3, pp. 289–301, 2008.

[30] L. Yang, B. Georgescu, Y. Zheng, Y. Wang, P. Meer, and D. Comaniciu, "Prediction based collaborative trackers (pct): A robust and accurate approach toward 3d medical object tracking," *IEEE Transactions on Medical Imaging*, vol. 30, no. 11, pp. 1921–1932, 2011.

[31] A. Qiu, L. Younes, and M. I. Miller, "Principal component based diffeomorphic surface mapping," *IEEE Transactions on Medical Imaging*, vol. 31, no. 2, pp. 302–311, 2011.

[32] C. Jud, F. Preiswerk, and P. C. Cattin, "Respiratory motion compensation with topology independent surrogates," in *Workshop on imaging and computer assistance in radiation therapy*, 2015.

[33] C. Jud, A. Giger, R. Sandkühler, and P. C. Cattin, "A localized statistical motion model as a reproducing kernel for non-rigid image registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 261–269.

[34] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3581–3589.

[35] J. Krebs, T. Mansi, N. Ayache, and H. Delingette, "Probabilistic motion modeling from medical image sequences: Application to cardiac cine-mri," in *Statistical Atlases and Computational Models of the Heart. Multi-Sequence CMR Segmentation, CRT-EPiggy and LV Full Quantification Challenges*. Cham: Springer, 2020, pp. 176–185.

[36] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[38] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[39] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.

[40] V. Fortuin, G. Rätsch, and S. Mandt, "Multivariate time series imputation with variational autoencoders," *arXiv preprint arXiv:1907.04155*, 2019.

[41] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "An unsupervised learning model for deformable medical image registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9252–9260.

[42] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.

[43] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.

[44] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2017–2025.

[45] F. Chollet, "Keras," https://keras.io, 2015.

[46] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[47] J. Duchi, "Derivations for linear algebra and optimization," *Berkeley, California*, vol. 3, no. 1, pp. 2325–5870, 2007.