Human Biophysics as Network Weights: Conditional Generative Models for Ultra-fast Simulation

Shihan Ma\textsuperscript{1,2†}, Alexander Kenneth Clarke\textsuperscript{1†}, Kostiantyn Maksymenko\textsuperscript{3}, Samuel Deslauriers-Gauthier\textsuperscript{3,4}, Xinjun Sheng\textsuperscript{2,5}, Xiangyang Zhu\textsuperscript{2,5} and Dario Farina\textsuperscript{1}

\textsuperscript{1}Department of Bioengineering, Imperial College London, London, UK.
\textsuperscript{2}State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University, Shanghai, China.
\textsuperscript{3}Neurodec, Sophia Antipolis, France.
\textsuperscript{4}Inria Centre at Université Côte d’Azur, Nice, France.
\textsuperscript{5}Meta Robotics Institute, Shanghai Jiao Tong University, Shanghai, China.
Correspondence to d.farina@imperial.ac.uk and mexyzhu@sjtu.edu.cn.

†These authors contributed equally to this work.

Abstract
Simulations of biophysical systems have provided a huge contribution to our fundamental understanding of human physiology and remain a central pillar for developments in medical devices and human machine interfaces. However, despite their successes, such simulations usually rely on highly computationally expensive numerical modelling, which is often inefficient to adapt to new simulation parameters. This limits their use in dynamic models of human behavior, for example in modelling the electric fields generated by muscles in a moving arm. We propose the alternative approach to use conditional generative models, which can learn complex relationships between the underlying generative conditions whilst remaining inexpensive to sample from. As a demonstration of this concept, we present BioMime, a hybrid architecture that
Learning Dynamic Human Biophysics with Generative Models combines elements of deep latent variable models and conditional adversarial training to construct a generative model that can both transform existing data samples to reflect new modelling assumptions and sample new data from a conditioned distribution. We demonstrate that BioMime can learn to accurately mimic a complex numerical model of human muscle biophysics and then use this knowledge to continuously sample from a dynamically changing system in real-time. We argue that transfer learning approaches with conditional generative models are a viable solution for dynamic simulation with any numerical model.

**Keywords:** Conditional generative model, transfer learning, biophysical simulation, neurophysiology, electromyography

1 Main

Biophysical simulations are a cornerstone of modern biomedical research and engineering, allowing initial exploration of experimental hypotheses and fast iteration of designs prior to physical implementation [1, 2]. Decades of continuous developments have seen such models go from a few equations, such as in Hodgkin and Huxley’s hugely impactful work on spiking neurons [3], to highly complex physics engines and large numerical models with thousands of individual parameters [4]. The rapid expansion in the complexity and fidelity of biophysical simulations has played a major role in advancing their corresponding domains [5], and has even generated entirely new avenues of investigation, such as neurophysiological source reconstruction and embodied artificial intelligence [6, 7].

Despite their successes, the increasing complexity of biophysical simulations has come with a corresponding increase in the associated computational burden, which often limits adoption [5]. Computational complexity is particularly problematic when simulating a dynamic event which rapidly changes the modelling conditions, for example the deformation of the volume conductor in a moving forearm [8], or the changes in mechanical response of the tendons in a bending knee [9]. Usually, the only solution to these types of problem is to discretise time, splitting one dynamic simulation into many individual static simulations, a task which is often computationally demanding or even unfeasible. A related issue is that adapting the model to new sets of parameters, such as a new individual, means that the model needs to be rerun with every parameter updated [10], which is a major barrier to the promised “digital twin” of personalised healthcare simulations [11].

We propose that a better approach to modelling dynamic human systems is to combine transfer learning approaches with conditional generative models [12], training a deep latent variable model with a numerical simulation’s outputs and then predicting the emissions of the interpolating system states. Such a model could then generate samples inexpensively for a variety of static simulations, or continually morph the emissions of a continuously evolving
dynamic state. To achieve these aims, we propose BioMime, a hybrid conditional model that fuses elements of the variational auto-encoder (VAE) and generative adversarial network (GAN) [13, 14]. BioMime can accurately learn the outputs of a computationally-expensive simulation, and continuously interpolate between different stages of the system, essentially converting a static model into a dynamic one.

![Deep latent variable model, BioMime](image)

**Fig. 1 Deep latent variable model, BioMime.** 

**a.** The generative architecture of the BioMime model, which consists of a conditional decoder that takes either an encoded emission $\tilde{x}$ or standard normal sample as a latent input $z$ and then outputs an emission $\tilde{x}$ that reflects the learned generative factors of the desired biophysical system state, the specified conditions $c_s$. The encoder is particularly useful when the objective is to conditionally modify emissions whilst retaining the generative effects of unspecified generative factors. 

**b.** BioMime is trained with an adversarial loss $L_{GAN}$ using a discriminator which seeks to distinguish between the generative emissions of the encoder-decoder generator $\tilde{x}$ and the emissions of the numerical simulator $x_0$. Like the decoder, the discriminator is also conditioned on $c_s$ when given samples from the generator or simulator, meaning the generator must learn a conditional implicit density. To further drive conditional behaviour in the discriminator, the simulation emissions are paired with specified conditions which are either correct $c_0$ or incorrect $c_r$. To allow for ab initio generation and to stabilise training, an additional Kullback-Leibler divergence $L_{KL}$ term is minimised between $z$ and a standard normal prior. Finally we found empirically that the addition of a cycle-consistency loss $L_{cyclic}$ improved training stability and gave an increase in model performance (Supplementary Fig. 4).

**c.** Rapid generation of a dynamically evolving emission ab initio. A sample is taken from the prior and then continuously transformed over time using a sweep of the specified conditions.

BioMime takes the form of a conditional deep latent variable model, with an encoder that takes the simulation emissions and learns a compressed latent representation (Fig. 1a). The latent representation is then concatenated with the linearly projected desired simulation parameters and is passed to the decoder, which reconstructs the new emission. This allows BioMime to accurately mimic a simulation’s internal state and then generate emissions that
replicate a dynamic change in that state. BioMime is trained adversarially using a conditional discriminator (Fig. 1b), as in conditional GANs [15, 16].

One additional feature of BioMime is that it can also generate realistic samples without using an encoded representation of a simulation emission. Similar to a VAE, this is achieved by adding a Kullback-Leibler (KL) divergence term to the loss, minimising the distance between the sufficient statistics extracted from the latent representation and a gaussian prior. Sampling from the prior negates the need to use an encoded emission for generation if retaining the unspecified generative factors of the input data is not required. This is useful for quickly constructing a dynamic simulation, of which the new emissions can be generated by sampling one latent representation from the prior distribution and reusing it with all the evolving simulation parameters (Fig. 1c). We also found empirically that the regularisation added by the KL divergence term, as well as an additional cycle-consistent term [17], dramatically improved model convergence.

**Fig. 2 Capturing a complex biophysical simulation.** The effectiveness of BioMime is validated on a simulation of surface electromyogram (sEMG) signals recorded from an arm, the myoelectric output of motor neuron activities from the spinal cord as detected by an array of electrodes placed on the skin above the muscles. Each spike from a specific motor neuron activates an associated pool of muscle fibres (the combination of the motor neuron and the muscle fibres it controls is called a motor unit), forming an electrical potential field that is filtered by the volume conductor and sensed at the surface electrodes as the motor unit action potential (MUAP) waveform. Simulated templates of MUAPs are the building blocks of simulated sEMG, where they are convolved with the desired spike activities and then summed (with optional added gaussian noise). For the emissions of the finite element method simulation used to train BioMime, the bulk of MUAP variance is explained by six specified generative factors (highlighted in blue).
2 Results

2.1 BioMime accurately mimics computationally expensive models

To demonstrate the ability of BioMime to capture a complex biophysical simulation, the model was trained on the emissions of a modern numerical simulation of volume conduction, used for generating surface electromyographic (EMG) signals (Fig. 2). Surface EMG is a typical example of neurophysiological time series signal, consisting of the linearly mixed contributions of a number of individual sources with repetitive emissions [18]. It provides non-invasive information on neural activity and is commonly used in medical engineering and physiology research [19–22]. However, whilst numerical models of surface EMG generation have become highly advanced, they incur a huge computational burden when simulating the dynamic contractions, as the internal states of the model need to be continuously recalculated. This has proven to be a hindrance to the development of inverse models which can manage dynamic changes in the muscle [23], such as signals from a flexing forearm. The advanced numerical model we used to validate BioMime uses a finite element method to simulate motor unit action potentials (MUAPs), the basic component of EMG signal [24]. The shape of each MUAP waveform is dependent on a number of factors, such as the location of the muscle fibres in the forearm (Fig. 3a).

BioMime was trained using a set of MUAP emissions from the simulator, with each waveform labelled with the specific conditions used to generate it. A held-out test set of emissions was used to validate the model, which was done by taking a MUAP emission from the numerical simulator and using BioMime to transform the emission conditioned on another set of parameters. The output of BioMime was then compared to the equivalent emission from the simulator. BioMime was able to consistently and accurately transform MUAPs based on the specified conditions, generating waveforms which were extremely similar in shape to the simulation equivalent (Fig. 3b). The mean normalised root mean square error (nRMSE) across all samples was 1.8% when evaluated on the held-out test set and there was equal performance across all six of the specified conditions (Fig. 3c).

An interesting result of conditioning the decoder is that the encoder outputs tended to lose the information about the specified conditions, analogous to semi-supervised disentanglement methods [25, 26]. We investigated this effect with an informativeness metric [27], using a non-linear regressor to inversely predict the specified conditions used to generate an emission from the encoded waveform. The latent features held very little information about the specified conditions, with a median informativeness score slightly above chance at 35.9%. We speculate that the disentangling of the generative effects of the specified conditions from those that were unspecified, i.e. not explicitly included during training, improves the predictability of the model response to changes in the specified conditions by reducing the burden on the decoder.
2.2 Predicting the emissions of a dynamically evolving system

After capturing the volume conduction effect from the numerical simulator, BioMime could conditionally transform MUAPs such that they accurately interpolated between those from the original simulation, converting the static simulation into a dynamic one with very little computational expense (Fig. 4a). Visualization using t-distributed stochastic neighbor embedding (t-SNE) dimensionality reduction [28] shows that the samples in the original dataset were grouped into small clusters (Fig. 4b), as the MUAPs in each muscle were highly similar. The waveforms generated by interpolating the specified conditions closely matched the distribution of the MUAPs from the original simulation. MUAPs generated by BioMime using specified conditions outside
of those in the training set generally remained close to the original distribution until the specified conditions had deviated by a relative difference greater than 60%.

The ability of BioMime to accurately interpolate between simulation emissions was further confirmed by building a set of sweeps of specified conditions that had three ground truth emissions from the numerical dataset at equal distances along the swept paths for all samples in the test set (Fig. 4c). As the specified conditions moved further away from the original values used to generate the waveform, the MUAP shape increasingly deviated away from the ground truth. However, the discrepancy was not large (nRMSE less than 2.0%) until the sweep intersected with the third ground truth, which was the sample with the generative factors most different from the original point (Methods 4.3). This means that even for situations where very high interpolation accuracies are desired during a dynamic traversal, a relatively small dataset from the computational expensive numerical simulator is sufficient.

**Fig. 4** BioMime can predictively interpolate and extrapolate between simulation conditions and beyond. *a*, A set of BioMime-transformed MUAPs sampled from a continuous sweep of specified conditions. As the specified conditions are continuously traversed, they will occasionally intersect with the generative factors used to generate a static emission from the simulator (displayed as superimposed black lines), which can be used as a ground truth. *b*, A t-SNE projection of the MUAP emissions from the original simulation (black), interpolated BioMime transformations (blue) and extrapolated BioMime transformations beyond the specified conditions in the training set with colour set by euclidean distance from these conditions. The distribution of the interpolated MUAPs closely matches those from the original simulation, whilst the extrapolated emissions only start to deviate at relative distances greater than 60% from the ranges of specified conditions in the training set. *c*, Mean normalised root mean squared error of the BioMime-transformed MUAPs compared to their simulated counterparts from the test set during a traversal. The original simulated MUAP is encoded and then continuously transformed away from its origin. This traversal is selected to move the MUAPs in the test set through three simulation ground truths, showing that the predictive error of BioMime increases as the specified conditions move further away from their origin. BioMime is able to compensate for large moves away from the original conditions before the error starts to rise.
2.3 Inexpensively simulating a dynamic human biophysical system

Once trained, BioMime is able to rapidly generate emissions using any variety of specified conditions. Synthesizing 3,200 MUAPs under 1,000 continuous conditions only takes around 80 seconds, which is two orders of magnitude faster than the state-of-the-art numerical model [24]. The low computational burden allows BioMime to produce continuous emissions of signals that capture the dynamics of the physical system. This is easily superior to traditional numerical methods, as adapting the finite element model to new parameters requires complex integration of several pre-processing steps, which is extremely time demanding and computationally expensive [8, 29].

To demonstrate the utility of BioMime as a method to convert a static biophysical simulation into a dynamic one, the trained model was used to simulate the sEMG signals from a dynamic contraction of the forearm [30–32]. This is prohibitively time consuming using numerical methods due to the computational cost of simulating hundreds of discrete changes to the model state. Latent samples were generated by encoding a total of 1,336 MUAPs from the four extensors and three flexors synthesized by the numerical simulator, which were then conditionally transformed across a sweep of specified conditions to match known anatomical changes (Fig. 5a). The library of transformed MUAPs for each MU could then be convolved with a physiologically accurate set of motor neuron spike trains representing two force ramps (Fig. 5b). Finally the individual MU sources for each electrode channel were summed to give the final sEMG activity (Fig. 5c).

3 Discussion

Humans are dynamic systems, and computationally-feasible methods are needed to better reflect that fact in biophysical simulations. By transferring the knowledge of numerical simulations to a conditional generative model, we demonstrate that the cost associated with simulating an evolving biophysical system can be largely mitigated without losing the prediction fidelity. With the correct architecture design, such models can double both as a rapid way to transform existing simulation emissions to reflect new system states and as de novo generators for new emissions.

An example of a conditional generative model, BioMime, was demonstrated to generalisably and accurately learn the output of a large number of differential equations used by a modern numerical simulation describing a complex biophysical system. It was further able to continuously interpolate between system states in a manner that allowed a rapid and robust generation of dynamic biophysical emissions that closely tracked those of the teacher finite element method-based simulation. In practice, the generation of the required emission set was two orders of magnitude faster than generating a corresponding set using the simulation. These efficiencies allowed us to demonstrate the first practical model for simulating the myoelectric output of a moving forearm.
Fig. 5 BioMime can be used to mimic the dynamic changes of a biophysical system. a, Two representative MUAPs generated by BioMime, which are continuously morphed using the predicted changes during a forearm flexion and extension as specified conditions. The upper MUAP is from an extensor and the bottom one is from a flexor. The dotted lines show the change of MUAP amplitudes with time. b, Raster plots of 180 simulated motor neurons in one representative extensor muscle of the forearm over two force ramps, with overlaid indicators of the predicted changes to motor unit depth and fibre length during the dynamic contraction. c, The simulated surface EMG signals at electrode channels on the flexor muscles (upper) and the extensor muscles (bottom). The deep blue lines show the excitation levels to the flexor muscle group (upper) and the extensor muscle group (bottom). This complex dynamic simulation is only possible due to the low computational cost of the hundreds of MUAP transformations conducted by BioMime.

In the field of neurophysiological signal decomposition, there is a clear need for simulations that capture the effect of dynamic changes to the volume conductor, which has so far proved a bottleneck to the development and validation of source separation algorithms that can operate on this highly non-stationary signal [23]. The proposed methodology goes some way in meeting such a demand.

The main disadvantage of using a conditional generative model for simulation is that it is reliant on the quality and quantity of the training emissions, although this is the case with any method of supervised or unsupervised machine learning. It is important to emphasise that the methods outlined in this paper are designed to augment and approximate, rather than replace, high-quality numerical modelling. In many biophysical domains, the dynamic changes in system parameters in response to some perturbations remain poorly characterised, which restricts the utility of BioMime for simulating such systems. This is particularly true when modelling dynamic changes in a volume conductor for generating sEMG signals, and whilst we attempted to carefully estimate the likely parameter changes during a forearm contraction, there is a dearth of such information in the literature. However, we hope that the promise
offered by conditional generative models in allowing such dynamic systems to be practically captured will further stimulate research in this direction.

There is no requirement that BioMime be trained on the emissions of only one simulation. A promising future research direction would be to use the proposed methodology to learn the emissions of multiple computational models that simulate the same system from different perspectives, for example synthesizing the simulation of electrical potentials generated by motor neuron activity with the complex human biomechanical system they stimulate during realistic movement [7]. The computational burden of such full-spectrum simulations of biophysical systems is massive, precluding the modelling of a range of movements, but this becomes possible by training conditional generative models to effectively merge the emissions of the component simulations.

Contemporary methods of biophysical simulation continue to offer a range of opportunities to both test physiological hypotheses and as a test bed for medical technologies. High representational-capacity generative modelling, such as the methods described in this paper, offers an enormously flexible opportunity for leveraging the domain knowledge contained within these complex systems. We anticipate that the pipeline of a back-end numerical model and a front-end generative model will become increasingly common in future simulation design, and we look forward to the corresponding expansion in dynamic simulation that these techniques will enable.

4 Methods

4.1 Model Architecture

The proposed deep latent variable model, BioMime, takes the form of a probabilistic autoencoder architecture with an encoder and decoder network (Supplementary Table 1). The encoder network consists of five convolutional layers, each of which includes a 3D convolution with kernel size 3 followed by a $1 \times 1$ convolution with a skip connection. A parametric rectified linear unit activation (PReLU) operation is performed after each convolution layer. The convolutional output is then flattened and passed to two linear layers, which estimate the sufficient statistics of the approximate posterior, mean and variance, respectively. During training, the approximate posterior is sampled by the reparameterisation trick. During inference, the expectation of the posterior is used as the latent representation (Supplementary Algorithm 2) for transforming the input samples, whilst for \textit{ab initio} generation, the latents are taken from a gaussian prior with zero mean and identity covariance (Supplementary Algorithm 3). The latent representation is then concatenated with a 64-dimensional learned linear projection of the six specified conditions before being passed to the decoder.

The decoder consists of four convolutions followed by two upscaling blocks, each consisting sequentially of a time-scaling module, a 3D convolution, and a $1 \times 1$ convolution. The time-scaling module is a mean pooling/unpooling bank with learned weights (Supplementary Fig. 1), specialized for dilating or
compressing the time sequences, which is essential in modelling physiological signals but computationally expensive to achieve through many convolutional operations. Inspired by the dynamic convolution in [33] and the multi-scale spatial pooling in [34], the time-scaling module uses a series of experts $e_k$ with different scaling factors to dilate or compress the inputs, conditioned on the specified conditions $c_s$. The contribution of each expert is modulated by a scalar weight, which is the output of a three-layer multilayer perceptron with the six specified conditions as input:

$$y = \sum_k \pi_k(c_s)e_k(x)$$

s.t. $0 \leq \pi_k(c_s) \leq 1, \sum_k \pi_k(c_s) = 1$ (1)

where $\pi_k(c_s)$ is the weight of the $k$th expert conditioned on the desired states $c_s$. In this study, we used eight experts with scaling factors linearly spaced from 0.25 to 2.0, which showed optimal performance in both convergence and reconstruction accuracy (Supplementary Fig. 4).

### 4.2 Training Pipeline

Whilst BioMime has a deep latent variable architecture similar to a VAE, it is not trained variationally. Instead we found that a conditional adversarial method gave the best performance. The output of BioMime is inspected by a conditional discriminator network of the similar architecture as the encoder, trained to identify whether samples are from the original dataset of simulated emissions or generated by BioMime. Conditioning was performed by concatenating the output of the first convolutional layer with the specified conditions.

The goals of training BioMime are to (1) produce novel data samples that are realistic enough to fool the discriminator and (2) enable generation by both sampling from a prior distribution and by encoding an existing emission. These are achieved by optimizing the following objective function:

$$\mathcal{L}_G = \lambda_1 \mathcal{L}_{GAN} + \lambda_2 \mathcal{L}_{KL} + \lambda_3 \mathcal{L}_{cyclic}$$ (2)

The first term $\mathcal{L}_{GAN}$ is an adversarial loss, which evaluates the performance of BioMime using the conditional discriminator. The second term $\mathcal{L}_{KL}$ is the Kullback-Leibler divergence ($\mathcal{D}_{KL}(\cdot\|\cdot)$) between the predicted distribution of the latent feature and the standard normal distribution $\mathcal{N}(0,1)$. Minimizing $\mathcal{L}_{KL}$ regularises the latent space to approach $\mathcal{N}(0,1)$. This enables the model to generate new data by sampling from the prior. An additional cycle-consistency loss $\mathcal{L}_{cyclic}$ is included to improve training stability and generation accuracy (Supplementary Fig. 4), which is the mean-squared error between the input sample and the reversed sample (Fig. 1b). In our experiments, the hyperparameters $\lambda_1$ was set 10, $\lambda_2$ was an annealing weight [35] increased from 0 to
0.05 in 30,000 iterations, and \( \lambda_3 \) was set 0.5. Comparisons of constant, logistic, and linear KL divergence schedules (\( \lambda_2 \)) are shown in Supplementary Fig. 4.

The objective of the discriminator is to differentiate the data samples produced by the generator from those in the original dataset of simulator emissions, whilst also detecting whether the specified conditions are likely to be those that the simulator used to generate a specific emission. This is done by minimizing:

\[
\mathcal{L}_D = - \mathbb{E}_x[\log D(x, c_s)] - \{ \mathbb{E}_x[\log(1 - D(x, c_r))] - \mathbb{E}_x[\log(1 - D(G(x, c_s), c_s))] \}/2
\]  

(3)

In this way, the discriminator only predicts an input as real when the sample \( x \) is realistic and matches the desired conditions \( c_s \). Real samples \( x \) with random conditions \( c_r \) or synthesized samples \( G(x, c_s) \) with matched conditions \( c_s \) are predicted as fake.

**Training Process** The ratio between the discriminator and the generator updates was set 1 : 1 (Supplementary Algorithm 1). The discriminator and the generator were trained for 45 epochs (405,000 iterations) with RMSprop optimizer and learning rate of \( 1 \times 10^{-5} \). The number of input signals in each training mini-batch was set to 32. Training took 120 hours (~5 days) on a NVIDIA RTX 2080Ti.

**4.3 Data Preparation**

We used an advanced finite element-based numerical model with a realistic forearm anatomy provided by the Neurodec software [24] to prepare the dataset of simulation emissions (Supplementary Methods and Supplementary Fig. 2). To simulate the motor unit action potentials with sufficient variation of conditions, a total of 1,500 motor units were randomly generated within the eight forearm muscles, including the superficial muscles in the flexors and extensors. Each motor unit controlled a bundle of muscle fibres that clustered within a sub-volume of the muscle (Fig. 2). Six factors of variations that explained the bulk of MUAPs were defined as the specified conditions, covering the properties of current sources, motor units, and the volume conductor.

For each MU, the action potential templates under 256 conditions were generated, which were the combinations of four fibre densities (200, 266, 333, 400 fibres per mm\(^2\)), four current source propagation velocities (3.0, 3.5, 4.0, 4.5 m/s), four neuromuscular junction positions (0.4, 0.46, 0.53, 0.6), and four fibre lengths in ratio (0.85, 0.95, 1.05, 1.15). The other two specified conditions are the depth and the medial-lateral position of the MU centre, which was defined as the geometric centre of all the muscle fibres it controlled. The six specified conditions were linearly normalized between 0.5 and 1. The influences of the six conditions on the MUAP attributes are visualized in Supplementary Fig. 3.
There were a total of 384,000 samples in the dataset. For each muscle, 75% of MUs were randomly selected as the training set (288,000 samples) and the others as the held-out test set (96,000 samples). The MUAP templates were simulated with a grid of $10 \times 32$ electrodes that covered the bellies of the forearm muscles, and were represented as matrices with size $10 \times 32 \times T$. The number of time samples $T$ varied among muscles due to the different fibre lengths and the current source propagation velocities. During the preprocessing of the MUAP data, the signals were first downsampled to 2,000 Hz and were centered with their accumulative power. Finally, the signals were cut to a common length of 96 temporal samples.

4.4 Model Validation

To validate the model, we evaluated the ability of BioMime to transform the existing MUAP templates to new sets of simulation parameters. This was performed by optimizing the model on the training dataset and evaluating its performance on the held-out test set. We randomly selected two data samples with the same unspecified generative factors and transformed the first sample to match the second sample’s specified conditions. The generation accuracy was evaluated by the normalized root mean square between the predicted MUAP and the ground truth MUAP.

4.5 Informatic Analyses

**Normalized r.m.s. error** To quantify the accuracy of the generated signals compared with the ground truth, the normalized root mean square error was used as a metric:

$$\text{nRMSE} = \frac{\sqrt{\sum_{h,w,t} (x_{h,w,t} - \hat{x}_{h,w,t})^2} / H/W/T}{x_{\text{max}} - x_{\text{min}}}$$

(4)

where $x$ and $\hat{x}$ indicate the ground truth MUAP from the numerical model and the synthesized MUAP by BioMime. The variables $h$, $w$, $t$ are the summation over the rows and columns of the electrode and the time samples, with the total number $H$, $W$, and $T$, respectively. $x_{\text{max}}$ and $x_{\text{min}}$ are the maximum and minimum values of the sample. The normalized root mean square error is often expressed as a percentage, with a lower value indicating a less residual variance for the model and a higher generation accuracy.

**t-distributed stochastic neighbor embedding (t-SNE)** To visualize the implicit structure of the MUAP data, we used t-SNE to project the signals to a low-dimensional submanifold. The scikit-learn t-SNE module was used with default settings (dimension of the embedded space 2, perplexity 30, learning rate 200, maximum number of iterations 1000, initialization of embedding using principle component analysis) [36]. The samples were visualized in the coordinates given by t-SNE, where similar samples were close together in the submanifold.
Learning Dynamic Human Biophysics with Generative Models

Informativeness score We explored the amount of information of the specified conditions embedded in the latent representations of the generator. The informativeness of the latent vector $z$ about the $i$th generative factor $c_{si}$ can be quantified by the prediction accuracy $P(c_{si}, f(z))$, where $P$ is an accuracy metric and $f$ is a regressor [27]. The informativeness depends on 1) the regressor’s capability to extract information from the latent representations and 2) the way the specified conditions are embedded in the latent features, e.g., disentanglement of the specified conditions in the representations will make the regression easier.

We used multilayer perceptron with non-linear activation functions to predict $c_{si}$ from $z$. The prediction accuracy was measured as the percentage of correctly predicted samples. A sample was regarded as correct when the relative error between the predicted and real conditions was within a threshold. For instance, with the threshold of 0.05, a sample with ground truth condition $a$ will be evaluated as correct if the predicted condition is between $0.95a \sim 1.05a$. We chose ten threshold levels, linearly spaced from 0.01 to 0.10, and calculated the average accuracy as the informativeness score (Supplementary Table 2).

5 Funding

This study is supported by the National Natural Science Foundation of China (Grant No. 91948302, 52175021), the European Research Council Synergy Grant NaturalBionicS (contract 810346), the EPSRC Transformative Healthcare, NISNEM Technology (EP/T020970), and the BBSRC, “Neural Commands for Fast Movements in the Primate Motor System” (NU-003743).

6 Code availability

All code was implemented in Python using the deep learning framework PyTorch. Code which implements the models used in this paper is available at https://github.com/shihan-ma/BioMime and is provided under the GNU General Public License v3.0.

References

[1] Gerstner, W., Sprekeler, H., Deco, G.: Theory and simulation in neuroscience. science 338(6103), 60–65 (2012)

[2] Halilaj, E., Rajagopal, A., Fiterau, M., Hicks, J.L., Hastie, T.J., Delp, S.L.: Machine learning in human movement biomechanics: Best practices, common pitfalls, and new opportunities. Journal of biomechanics 81, 1–11 (2018)

[3] Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of physiology 117(4), 500 (1952)
[4] Breakspear, M.: Dynamic models of large-scale brain activity. Nature neuroscience 20(3), 340–352 (2017)

[5] Baby, D., Van Den Broucke, A., Verhulst, S.: A convolutional neural-network model of human cochlear mechanics and filter tuning for real-time applications. Nature machine intelligence 3(2), 134–143 (2021)

[6] Fuchs, M., Drenckhahn, R., Wischmann, H., Wagner, M.: An improved boundary element method for realistic volume-conductor modeling. IEEE Transactions on biomedical engineering 45(8), 980–997 (1998)

[7] Caggiano, V., Wang, H., Durandau, G., Sartori, M., Kumar, V.: Myosuite: A contact-rich simulation suite for musculoskeletal motor control. In: Learning for Dynamics and Control Conference, pp. 492–507 (2022). PMLR

[8] Pereira Botelho, D., Curran, K., Lowery, M.M.: Anatomically accurate model of emg during index finger flexion and abduction derived from diffusion tensor imaging. PLoS computational biology 15(8), 1007267 (2019)

[9] Erdemir, A.: Open knee: open source modeling and simulation in knee biomechanics. The journal of knee surgery 29(02), 107–116 (2016)

[10] Numata, S., Itatani, K., Kanda, K., Doi, K., Yamazaki, S., Morimoto, K., Manabe, K., Ikemoto, K., Yaku, H.: Blood flow analysis of the aortic arch using computational fluid dynamics. European Journal of Cardio-Thoracic Surgery 49(6), 1578–1585 (2016)

[11] Kamel Boulos, M.N., Zhang, P.: Digital twins: from personalised medicine to precision public health. Journal of Personalized Medicine 11(8), 745 (2021)

[12] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q.: A comprehensive survey on transfer learning. Proceedings of the IEEE 109(1), 43–76 (2020)

[13] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)

[14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems 27 (2014)

[15] Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2image: Conditional image generation from visual attributes. In: European Conference on Computer Vision, pp. 776–791 (2016). Springer
Learning Dynamic Human Biophysics with Generative Models

[16] Gebauer, N.W., Gastegger, M., Hessmann, S.S., Müller, K.-R., Schütz, K.T.: Inverse design of 3d molecular structures with conditional generative neural networks. Nature communications 13(1), 1–11 (2022)

[17] Zhu, J.-Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232 (2017)

[18] Merletti, R., Farina, D.: Surface Electromyography: Physiology, Engineering, and Applications. John Wiley & Sons, ??? (2016)

[19] Farina, D., Jiang, N., Rehbaum, H., Holobar, A., Graimann, B., Dietl, H., Aszmann, O.C.: The extraction of neural information from the surface emg for the control of upper-limb prostheses: emerging avenues and challenges. IEEE Transactions on Neural Systems and Rehabilitation Engineering 22(4), 797–809 (2014)

[20] Ruonala, V., Meigal, A., Rissanen, S., Airaksinen, O., Kankaanpää, M., Karjalainen, P.: Emg signal morphology and kinematic parameters in essential tremor and parkinson’s disease patients. Journal of Electromyography and Kinesiology 24(2), 300–306 (2014)

[21] Saponas, T.S., Tan, D.S., Morris, D., Balakrishnan, R.: Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 515–524 (2008)

[22] Xiong, D., Zhang, D., Zhao, X., Zhao, Y.: Deep learning for emg-based human-machine interaction: a review. IEEE/CAA Journal of Automatica Sinica 8(3), 512–533 (2021)

[23] Glaser, V., Holobar, A.: Motor unit identification from high-density surface electromyograms in repeated dynamic muscle contractions. IEEE Transactions on Neural Systems and Rehabilitation Engineering 27(1), 66–75 (2018)

[24] Maksymenko, K., Clarke, A.K., Guerra, I.M., Deslauriers-Gauthier, S., Farina, D.: Towards the myoelectric digital twin: Ultra fast and realistic modelling for deep learning. bioRxiv, 2021–06 (2022)

[25] Gyawali, P.K., Li, Z., Ghimire, S., Wang, L.: Semi-supervised learning by disentangling and self-ensembling over stochastic latent space. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 766–774 (2019). Springer
[26] Wang, T., Yue, Z., Huang, J., Sun, Q., Zhang, H.: Self-supervised learning disentangled group representation as feature. Advances in Neural Information Processing Systems 34, 18225–18240 (2021)

[27] Eastwood, C., Williams, C.K.: A framework for the quantitative evaluation of disentangled representations. In: International Conference on Learning Representations (2018)

[28] Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research 9(11) (2008)

[29] Lowery, M.M., Stoykov, N.S., Dewald, J.P., Kuiken, T.A.: Volume conduction in an anatomically based surface emg model. IEEE Transactions on Biomedical Engineering 51(12), 2138–2147 (2004)

[30] Narici, M.V., Binzoni, T., Hiltbrand, E., Fasel, J., Terrier, F., Cerretelli, P.: In vivo human gastrocnemius architecture with changing joint angle at rest and during graded isometric contraction. The Journal of physiology 496(1), 287–297 (1996)

[31] Ito, M., Kawakami, Y., Ichinose, Y., Fukashiro, S., Fukunaga, T.: Non-isometric behavior of fascicles during isometric contractions of a human muscle. Journal of applied physiology (1998)

[32] Hodges, P., Pengel, L., Herbert, R., Gandevia, S.: Measurement of muscle contraction with ultrasound imaging. Muscle & Nerve: Official Journal of the American Association of Electrodiagnostic Medicine 27(6), 682–692 (2003)

[33] Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., Liu, Z.: Dynamic convolution: Attention over convolution kernels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11030–11039 (2020)

[34] Chen, J.-W., Wang, R., Ding, F., Liu, B., Jiao, L., Zhang, J.: A convolutional neural network with parallel multi-scale spatial pooling to detect temporal changes in sar images. Remote Sensing 12(10), 1619 (2020)

[35] Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. arXiv preprint arXiv:1511.06349 (2015)

[36] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
Human Biophysics as Network Weights: Conditional Generative Models for Ultra-fast Simulation

Shihan Ma\textsuperscript{1,2†}, Alexander Kenneth Clarke\textsuperscript{1†}, Kostiantyn Maksymenko\textsuperscript{3}, Samuel Deslauriers-Gauthier\textsuperscript{3,4}, Xinjun Sheng\textsuperscript{2,5}, Xiangyang Zhu\textsuperscript{2,5} and Dario Farina\textsuperscript{1}

\textsuperscript{1}Department of Bioengineering, Imperial College London, London, UK.
\textsuperscript{2}State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University, Shanghai, China.
\textsuperscript{3}Neurodec, Sophia Antipolis, France.
\textsuperscript{4}Inria Centre at Université Côte d’Azur, Nice, France.
\textsuperscript{5}Meta Robotics Institute, Shanghai Jiao Tong University, Shanghai, China.

Correspondence to d.farina@imperial.ac.uk and mexyzhu@sjtu.edu.cn.

\textsuperscript{†}These authors contributed equally to this work.

\textbf{Contents}

Supplementary Methods 1
Supplementary Algorithms 4
Supplementary Figures 6
Supplementary Tables 10
Supplementary References 12
Supplementary Methods

BioMime Model

Time-scaling Module

BioMime has an encoder-decoder structure. The encoder embeds the input samples into latent features while the decoder converts the latents to the output signals conditioned on the desired system state. We found empirically that multiple layers of fractionally-strided convolutions in the decoder, which are commonly used in restoring images, could not accurately generate MUAP signals that match the expected system conditions (Supplementary Figure 4). One possible reason is that simulating physiological signals requires the model to flexibly transform the long time sequence while traditional convolution filters have limitations when dealing with long distance relationships [1].

Inspired by the dynamic convolution in [2] and the multi-scale spatial pooling in [3], we proposed a time-scaling module, which is a temporal pooling/unpooling bank with learned weights specialized for dilating or compressing the time sequences in a large scale. A series of experts with different scaling factors are incorporated to endow the model with a dynamic scaling ability (Supplementary Figure 1). The weights of the experts are projected from the specified conditions by a feedforward neural network (three linear layers with two activation layers) and are normalized by softmax, as the scale of the expected MUAP greatly relies on the specified conditions, e.g., deeper location of fibres within the muscle results in more attenuated signals.

We found that ranging the temporal scaling factors between 0.25 and 2.0 is sufficient for well transforming the MUAP signals of the forearm muscles. Increasing the number of experts introduces more scaling factors but also increases the training time. Time-scaling module with eight experts generated accurate signals within an acceptable training iterations (Supplementary Figure 4). Therefore, eight experts were used in BioMime model.

Schedules of Kullback-Leibler (KL) divergence term

When trained with constant KL divergence term, the network converged slowly and the generation accuracy was low. We suppose this is because the regularisation of the latent space was strict, and thus the model was slow to capture the latent representations from the beginning of the training. We found that an annealing schedule for KL divergence loss [4], no matter linear or logistic, drove the model to converge faster and achieve higher prediction accuracy than with a constant weight (Supplementary Figure 4). By starting from a small coefficient, the model focused on extracting the information from the input data and reconstructing the output, rather than regularising the latent space that might interfere the learning process. As the weight increased, the model gradually regulated the latent features and reached a balance between the regularised space and the real space. The logistic and linear terms are formulated as follows:
\[ w = \begin{cases} \frac{1}{1 + e^{-k(x-x_0)}}, & \text{w in logistic form} \\ \frac{x}{N}, & \text{w in linear form} \end{cases} \] (1)

where \( x \) denotes the current iteration and the hyperparameters \( k = 1, \ x_0 = 6, \) and \( N = 30,000. \)

**Numerical simulation**

**Specified conditions**

The MUAP dataset consists of MUAP templates from eight forearm muscles generated by an advanced and realistic numerical model [5], as shown in Supplementary Figure 2(a). Both superficial flexors and extensors are included, i.e., Extensor carpi radialis bevis, Extensor carpi radialis longus, Palmaris longus, Flexor carpi ulnaris ulnar head, Flexor carpi ulnaris humeral head, Extensor carpi ulnaris, Extensor digitorum, Flexor digitorum superficialis, with 186, 204, 164, 205, 217, 180, 186, and 158 unique motor units (MUs), respectively. In total, we have 1,500 MUs in the dataset.

In each muscle, the fibres that belong to a single MU cluster within a sub-volume of the muscle. Here we define the centre of a MU as the centroid of all the points of intersections between the fibres and the reference plane. The 2D location of a MU centre is described by its depth and medial-lateral position (Supplementary Figure 2(b)). The depth of a MU is defined as the closest distance between the MU centre and the skin surface. The medial-lateral position of the MU is represented by the radian of the MU centre with respect to the reference electrode, $\widehat{AB}/p$, where \( p \) is the perimeter of the cross-section.

For each MU, the action potential templates were generated under 256 conditions, which are the combinations of four fibre densities (200, 266, 333, 400 fibres per \( \text{mm}^2 \)), four current source propagation velocities (3.0, 3.5, 4.0, 4.5 \( \text{m/s} \)), four neuromuscular junction positions (0.4, 0.46, 0.53, 0.6), and four fibre lengths in ratio (0.85, 0.95, 1.05, 1.15). Consequently, there were a total of six well-defined conditions that can be manually modified. These conditions were linearly normalized between 0.5 and 1. The influences of the six conditions on the MUAPs are visualized in Supplementary Figure 3, by generating MUAP templates using a cylindrical numerical model while individually changing each of the six conditions. We used the cylindrical model here as the 2D locations of the muscle fibres are difficult to change manually in the realistic numerical model. Note that rather than directly using the fibre density as one of the conditions, we used the number of muscle fibres in each MU, such that this condition changed more continuously.
Supplementary Algorithms

**Algorithm 1** Training procedure of BioMime

**Input:** MUAP samples labelled with their specified conditions \( \{x_0, c_0\} \), the desired conditions \( c_s \), randomly sampled conditions \( c_r \), number of training epochs \( S \), hyperparameters \( \lambda_1 = 10.0 \) for \( L_{GAN} \), \textit{anneal} \textit{func} for linearly increasing the weight \( \lambda_2 \) of \( L_{KL} \), \( \lambda_3 = 0.5 \) for \( L_{cyclic} \)

1: \( \theta_G, \theta_D \leftarrow \) initialize network parameters  
2: \textbf{for} \( n = 1 \) to \( S \) \textbf{do}  
3: \( \rho_r \leftarrow D(x_0, c_0) \)  \# real sample + real condition  
4: \( \rho_{f1} \leftarrow D(G(x_0, c_s), c_s) \) \# fake sample + real condition  
5: \( \rho_{f2} \leftarrow D(x_0, c_r) \) \# real sample + random condition  
6: \( L_D \leftarrow 0.1 \ast ((1 - \rho_r)^2 + (\rho_{f1}^2 + \rho_{f2}^2)/2) \)  
7: \( \theta_D \leftarrow \theta_D - \alpha \nabla_{\theta_D} L_D \) \# update discriminator  
8: \( \hat{x}, \mu, \log(\sigma^2) \leftarrow G(x_0, c_s) \) \# generate by morphing \  
9: \( \hat{x} \leftarrow G(\hat{x}, c_0) \) \# reverse generation  
10: \( \rho \leftarrow D(\hat{x}, c_0) \) \  
11: \( L_{GAN} \leftarrow (1 - \rho)^2 \)  
12: \( L_{cyclic} \leftarrow \|x_s - \hat{x}\|^2 \)  
13: \( L_{KL} \leftarrow -(1 + \log(\sigma^2) - \mu^2 - \sigma^2)/2 \)  
14: \( \lambda_2 \leftarrow \text{anneal\_func}(n) \)  
15: \( L_G \leftarrow \lambda_1 L_{GAN} + \lambda_2 L_{KL} + \lambda_3 L_{cyclic} \)  
16: \( \theta_G \leftarrow \theta_G - \alpha \nabla_{\theta_G} L_G \) \# update generator  
17: \textbf{end for}
Algorithm 2 Generating by morphing

Input: Desired specified condition $c_s$ and input sample $x$
Output: Simulated MUAPs conditioned on the desired system states $\tilde{x}$
1: $\mu, \sigma \leftarrow \text{Encoder}(x)$
2: $z \leftarrow \mu$
3: $\tilde{x} \leftarrow \text{Decoder}(z, c_s)$
4: Return $\tilde{x}$

Algorithm 3 Generating by sampling

Input: Desired specified condition $c_s$
Output: Simulated MUAPs conditioned on the desired system states $\tilde{x}$
1: $z \sim \mathcal{N}(0, 1)$
2: $\tilde{x} \leftarrow \text{Decoder}(z, c_s)$
3: Return $\tilde{x}$
Supplementary Figures

Fig. 1: Time-scaling module in decoder. The proposed time-scaling module is a temporal pooling/unpooling bank with learned weights. The weights are conditioned on the desired specified conditions by a feedforward layer. This module is specialized for dilating or compressing signals in time domain.
Fig. 2: | Illustration of realistic numerical model. a, The eight superficial muscles are visualized in the realistic forearm model. The radius and ulnar are shown in grey volumes. Electrode grid with $10 \times 32$ channels covers the muscle belly of the most forearm muscles. b, Definitions of the depth and medial-lateral position of the MU centre. MU centre is defined as the centroid of all the intersection points between the fibres in this MU and the reference plane. The closest distance between the MU centre and the skin surface is regarded as the depth of the MU. The medial-lateral position of the MU is represented by the radian of the MU centre with respect to the reference electrode, $\overline{AB}/p$, where $p$ is the perimeter of the cross section.
Fig. 3: Variations of spatial-temporal attributes of MUAP with changes of
(a) fibre density, (b) current propagation velocity, (c) fibre length, (d) neuromuscular junction, (e) depth of MU centre, and (f) medial-lateral position of MU centre. Color from deep to light denotes the variation of the normalized conditions from 0.5 to 1.0. Note, here, the MUAPs were generated by a numerical model with a cylindrical volume conductor to show clear variations of the MUAPs with the conditions. (a) Fibre density mainly changes the amplitude of the waveforms. (b) Current source propagation velocity scales the signal in the temporal domain. (c) Fibre length primarily modifies the duration of the signal. (d) The impact of neuromuscular junction is much more tricky. It moves the peak of the signals in the longitudinal direction and modifies the non-propagating component of the waveform [6]. (e) The depth of MU centre changes the low-pass filtering effect, such that the amplitude and the spatial distribution are changed. (f) The medial-lateral position of MU mainly alters the spatial distribution of MUAP traversely.
Fig. 4: | Boxplot of accuracy when changing schedules of KL divergence terms, increasing number of experts in time-scaling module, and including cycle-consistency loss or not. The accuracy was measured in the normalized root mean square error between the generated MUAP signals and the ground truth signals and was averaged among the held-out test set. The one-way Welch’s ANOVA of the generation accuracy on the KL divergence terms revealed a statistically significant main effect ($p < 0.001$). Post hoc comparisons using the Games-Howell post hoc procedure indicated that applying logistic or linear annealing schedule significantly improved the generation accuracy than using the constant schedule ($p < 0.001$). There was also a statistically significant difference between the means of accuracy with the number of experts (Welch’s one-way ANOVA, $p < 0.001$). The Games-Howell post hoc showed that the mean difference between the three levels all reached significance ($p < 0.001$). A Wilcoxon signed-rank test showed that including the cycle-consistency loss significantly improved the generation accuracy ($Z = -17.143, p < 0.001$).
### Supplementary Tables

**Table 1:** The network structure of encoder and discriminator in BioMime.

| Layer       | Parameters | Output       | Layer       | Parameters | Output       |
|-------------|------------|--------------|-------------|------------|--------------|
| Conv3d      | 3, 2, 1    | [16, 48, 5, 16] | Conv3d      | 3, 1, 1    | [16, 48, 5, 16] |
| Conv3d      | 3, 2, 1    | [32, 24, 3, 8]  | Conv3d      | 3, 2, 1    | [32, 24, 3, 8]  |
| Conv3d      | 3, 2, 1    | [64, 12, 2, 4]  | Conv3d      | 3, 2, 1    | [64, 12, 2, 4]  |
| Conv3d      | 3, (2, 1, 1), 1 | [128, 6, 2, 4] | Conv3d      | 3, (2, 1, 1), 1 | [128, 6, 2, 4] |
| Conv3d      | 3, 1, 1    | [256, 6, 2, 4]  | Conv3d      | 3, 1, 1    | [256, 6, 2, 4]  |
| Flatten     |            | [256 * 6 * 2 * 4] | AvgPool3d   | (6, 2, 4)  | [256, 1, 1, 1] |
| Linear ($\mu$) | 12288 $\to$ 16 | [16]    | Conv3d      | 1, 1, 0    | [1, 1, 1, 1]   |
| Linear ($\sigma$) | 12288 $\to$ 16 | [16]    | Squeeze     |            | [1]            |

* Each Conv3d in the encoder is followed by a PReLU layer. Each Conv3d in the discriminator is followed by a LeakyReLU with negative slope equal to 0.03.
The six specified conditions from $c_s^1$ to $c_s^6$ are number of muscle fibres, depth of MU centre, medial-lateral position of MU centre, position of neuromuscular junction, current propagation velocity, and fibre length, respectively.

Non-linear regressors with two to six hidden layers (hidden dimension 256) were trained.

One possible reason that the informativeness score suddenly decreased when the number of hidden layers in the regressor reached six is that the model overfitted the training dataset.

**Table 2:** Informativeness score (%) of indicating specified conditions $c_s$ from the latent features $z$.

| Hidden layers | $c_s^1$ | $c_s^2$ | $c_s^3$ | $c_s^4$ | $c_s^5$ | $c_s^6$ |
|---------------|--------|--------|--------|--------|--------|--------|
| 2 layers      | 50.85  | 37.87  | 37.89  | 32.99  | 24.10  | 14.06  |
| 3 layers      | 50.43  | 31.98  | 53.59  | 30.71  | 19.47  | 11.58  |
| 4 layers      | 49.75  | 36.17  | 53.18  | 29.89  | 23.54  | 11.36  |
| 5 layers      | 48.09  | 34.01  | 51.51  | 34.02  | 26.23  | 13.57  |
| 6 layers      | 16.13  | 16.13  | 19.10  | 13.60  | 5.00   | 7.50   |

1 The six specified conditions from $c_s^1$ to $c_s^6$ are number of muscle fibres, depth of MU centre, medial-lateral position of MU centre, position of neuromuscular junction, current propagation velocity, and fibre length, respectively.

2 Non-linear regressors with two to six hidden layers (hidden dimension 256) were trained.

3 One possible reason that the informativeness score suddenly decreased when the number of hidden layers in the regressor reached six is that the model overfitted the training dataset.
Supplementary References

References

[1] 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.

[2] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, “Dynamic convolution: Attention over convolution kernels,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11 030–11 039.

[3] J.-W. Chen, R. Wang, F. Ding, B. Liu, L. Jiao, and J. Zhang, “A convolutional neural network with parallel multi-scale spatial pooling to detect temporal changes in sar images,” Remote Sensing, vol. 12, no. 10, p. 1619, 2020.

[4] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, “Generating sentences from a continuous space,” arXiv preprint arXiv:1511.06349, 2015.

[5] K. Maksymenko, A. K. Clarke, I. M. Guerra, S. Deslauriers-Gauthier, and D. Farina, “Towards the myoelectric digital twin: Ultra fast and realistic modelling for deep learning,” bioRxiv, pp. 2021–06, 2022.

[6] R. Merletti and S. Muceli, “Tutorial. surface emg detection in space and time: Best practices,” Journal of Electromyography and Kinesiology, vol. 49, p. 102363, 2019.