MoGlow: Probabilistic and Controllable Motion Synthesis Using Normalising Flows

GUSTAV EJE HENTER*, SIMON ALEXANDERSON*, and JONAS BESKOW, Division of Speech, Music and Hearing, KTH Royal Institute of Technology, Sweden

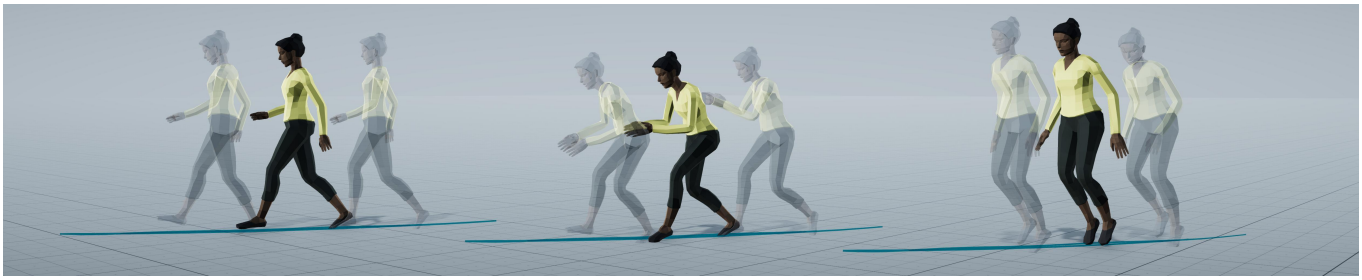

Fig. 1. Probabilistic motion generation. Random samples from our method can give many distinct output motions even if the input signal is the same.

Data-driven modelling and synthesis of motion is an active research area with applications that include animation, games, and social robotics. This paper introduces a new class of probabilistic, generative, and controllable motion-data models based on normalising flows. Models of this kind can describe highly complex distributions, yet can be trained efficiently using exact maximum likelihood, unlike GANs or VAEs. Our proposed model is autoregressive and uses LSTMs to enable arbitrarily long time-dependencies. Importantly, is is also causal, meaning that each pose in the output sequence is generated without access to poses or control inputs from future time steps; this absence of algorithmic latency is important for interactive applications with real-time motion control. The approach can in principle be applied to any type of motion since it does not make restrictive, task-specific assumptions regarding the motion or the character morphology. We evaluate the models on motion-capture datasets of human and quadruped locomotion. Objective and subjective results show that randomly-sampled motion from the proposed method outperforms task-agnostic baselines and attains a motion quality close to recorded motion capture.

CCS Concepts: • **Computing methodologies** → **Animation**; *Neural networks*; *Motion capture*.

Additional Key Words and Phrases: Generative models, machine learning, normalising flows, Glow, footstep analysis, data dropout

ACM Reference Format:
Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. 2020. MoGlow: Probabilistic and Controllable Motion Synthesis Using Normalising Flows. *ACM Trans. Graph.* 39, 4, Article 236 (July 2020), 14 pages. https://doi.org/10.1145/3414685.3417836

*Gustav Eje Henter and Simon Alexanderson contributed equally and are joint first authors.

Authors' address: Gustav Eje Henter, ghe@kth.se; Simon Alexanderson, simonal@kth.se; Jonas Beskow, beskow@kth.se, Division of Speech, Music and Hearing, KTH Royal Institute of Technology, Stockholm, Sweden.

0730-0301/2020/7-ART236
https://doi.org/10.1145/3414685.3417836

1 INTRODUCTION

A recurring problem in fields such as computer animation, video games, and artificial agents is how to generate convincing motion conditioned on high-level, "weak" control parameters. Video-game characters, for example, should be able to display a wide range of motions controlled by game-pad inputs, and embodied agents should generate complex non-verbal behaviours based on, e.g., semantic and prosodic cues. The advent of deep learning and the growing availability of large motion-capture databases have increased the interest in data-driven, statistical models for generating motion. Given that the control signal is weak, a fundamental challenge for such models is to handle the large variation of possible outputs – the limbs of a real person walking the same path twice will always follow different trajectories. Deterministic models of motion, which return a single predicted motion, suffer from regression to the mean pose and produce artefacts like foot sliding in the case of gait. They also lack motion diversity, leading to repetitive and non-engaging characters in applications. Taken together, we are led to conclude that for motion generated from the model to be perceived as realistic, it *cannot* be completely deterministic, but the model should instead generate *different* motions upon each subsequent invocation, given the same control signal. In other words, a stochastic model is required. Furthermore, *real-time* interactive systems such as video games require models with the lowest possible latency.

This paper introduces MoGlow, a novel autoregressive architecture for generating motion-data sequences based on normalising flows [Deco and Brauer 1994; Dinh et al. 2015, 2017; Huang et al. 2018; Kingma and Dhariwal 2018]. This new modelling paradigm has the following principal advantages:

(1) It is *probabilistic*, meaning that it endeavours to describe not just one motion, but *all* possible motions, and how likely each possibility is. Plausible motion samples can then be generated also in the absence of conclusive control-signal input (Fig. 1).
(2) It uses an *implicit model structure* [Mohamed and Lakshminarayanan 2016] to parameterise distributions. This makes it fast to sample from without assuming that observed values

follow restrictive, low-degree-of-freedom parametric families such as Gaussians or their mixtures, as done in, e.g., Fragkiadaki et al. [2015]; Uria et al. [2015].

(3) It allows exact and tractable probability computation, unlike variational autoencoders (VAEs) [Kingma and Welling 2014; Rezende et al. 2014], and can be *trained to maximise likelihood directly*, unlike generative adversarial networks (GANs) [Goodfellow 2016; Goodfellow et al. 2014].
(4) It is *task-agnostic* – that is, it does not rely on restrictive, situational assumptions such as characters being bipedal or motion being quasi-periodic (unlike, e.g., Holden et al. [2017]).
(5) It generates output sequentially and permits control schemes for the output motion with *no algorithmic latency*.
(6) It is capable of generating *high-quality motion* both in objective terms and as judged by human observers.

To the best of our knowledge, our proposal is the first motion model based on normalising flows. We evaluate our method on locomotion synthesis for two radically different morphologies – humans and dogs – since locomotion makes it easy to quantify artefacts and spot poor adherence to the control. A video presentation of our work with generated motion examples is enclosed in the supplement.

2 BACKGROUND AND PRIOR WORK

Mathematically, motion generation requires creating a sequence of poses from control input. We here review (Sec. 2.1) probabilistic machine-learning models of sequences, and then describe (Secs. 2.2 and 2.3) prior work on machine learning for motion synthesis.

2.1 Probabilistic generative sequence models

Probabilistic sequence models for continuous-valued data have a long history, with linear autoregressive models being an early example [Yule 1927]. Model flexibility improved with the introduction of hidden-state models like HMMs [Rabiner 1989] and Kalman filters [Welch and Bishop 1995], both of which still allow efficient probability computation (*inference*). Deep learning extended autoregressive models of continuous-valued data further by enabling highly nonlinear dependencies on previous observations, for example Fragkiadaki et al. [2015]; Graves [2013]; Uria et al. [2015]; Zen and Senior [2014], as well as nonlinear (continuous-valued) hidden-state evolution through recurrent neural networks, e.g., Hochreiter and Schmidhuber [1997]. All of these model classes have been extensively applied to sequence-modelling tasks, but have consistently failed to produce high-quality random samples for complicated data such as motion and speech. We attribute this shortcoming to the explicit distributional assumptions (e.g., Gaussianity) common to all these models – real data, e.g., motion capture, is seldom Gaussian.

Three methods for relaxing the above distributional constraints have gained recent interest. The first is to quantise the data and then fit a discrete model to it. Deep autoregressive models on quantised data, such as Kalchbrenner et al. [2018]; Salimans et al. [2017]; van den Oord et al. [2016, 2017]; Wang et al. [2018], are the state of the art in many low-dimensional (R^3 or less) sequence-modelling problems. However, it is not clear if these approaches scale up to motion data, with 50 or more dimensions. Quantisation may also introduce perceptual artefacts. A second approach is variational autoencoders [Kingma and Welling 2014; Rezende et al. 2014], which optimise a variational lower bound on model likelihood while simultaneously learning to perform approximate inference. The gap between the true maximum likelihood and that achieved by VAEs has been found to be significant [Cremer et al. 2018].

The third approach is GANs [Goodfellow 2016; Goodfellow et al. 2014], that generate samples from complicated distributions *implicitly*, by passing simple random noise through a nonlinear neural network. As GAN architectures do not allow inference, they are instead trained via a game against an adversary. GANs have produced some very impressive results in image generation [Brock et al. 2019], illustrating the power of implicit sample generation, but their optimisation is fraught with difficulty [Lucic et al. 2018; Mescheder et al. 2018]. GAN output quality usually improves by artificially reducing the generator entropy during sampling, compared to sampling from the distribution actually learned from the data, cf. Brock et al. [2019]. This is often referred to as "reducing the temperature".

While VAEs in principle have a partially-implicit generator structure, an issue dubbed "posterior collapse" means that VAEs with *strong decoders*, that can represent highly flexible distributions given the latent variable, tend to learn models where latent variables have little impact on the output distribution [Chen et al. 2017; Huszár 2017; Rubenstein 2019]. This largely nullifies the benefits of the implicit parts of the generator, leading to blurry and noisy output.

This article considers a less explored methodology called normalising flows [Deco and Brauer 1994; Dinh et al. 2015, 2017; Huang et al. 2018] (no relation to optical flow), especially a variant called Glow [Kingma and Dhariwal 2018], which, like GANs and quantisation, gained attention for highly realistic-looking image samples. We believe normalising flows offer the best of both worlds, combining a basis in likelihood maximisation and efficient inference like VAEs with purely implicit generator structures like GANs. Consequently, our paper presents one of the first Glow-based sequence models, and the first to our knowledge to combine autoregression and control, as well as to integrate long memory via a hidden state. The most closely-related methods are WaveGlow [Prenger et al. 2019] and FloWaveNet [Kim et al. 2019] for audio waveforms and VideoFlow [Kumar et al. 2020] for video. We extend these in several novel directions: Unlike Kim et al. [2019]; Prenger et al. [2019], our architecture is autoregressive ("closed-loop"), avoiding costly dilated convolutions and continuity issues (e.g., blocking artefacts) common in open-loop systems, cf. Juvela et al. [2019]. Unlike Kumar et al. [2020], our architecture permits output control. In contrast to all three models, we add a recurrent hidden state to enable long memory, which significantly improves the model. We also consider data dropout to increase adherence to the control signal.

2.2 Deterministic data-driven motion synthesis

While traditional motion synthesis uses concatenative approaches such as *motion graphs* [Arikan and Forsyth 2002; Kovar and Gleicher 2004; Kovar et al. 2002], there has been a strong trend towards statistical approaches. These can roughly be categorised into deterministic and probabilistic methods. Deterministic methods yield a single prediction for a given scenario, whereas probabilistic methods attempt

to describe a range of possible motions. Deterministically predicted pose sequences usually quickly regress towards the mean pose, cf. Ferstl et al. [2019]; Fragkiadaki et al. [2015], since that is the a-priori (i.e., no-information) minimiser of the MSE. Such methods thus require additional information to disambiguate pose predictions. Sometimes adding an external control signal suffices – lip motion is for example highly predictable from speech and has been successfully modelled with deterministic methods [Karras et al. 2017; Suwajanakorn et al. 2017; Taylor et al. 2017]. Locomotion generation represents a more challenging task, where path-based motion control does not suffice to unambiguously define the overall motion, and simple MSE minimisation results in characters that "float" along the control path. Proposals to overcome this issue in deterministic models include learning and predicting foot contacts [Holden et al. 2016], or the phase [Holden et al. 2017] or pace [Pavllo et al. 2018] of the gait cycle. Starke et al. [2020] generalised the idea of motion phase to complex motion by letting each bone in a character follow a separate motion phase. Autoregressively feeding in previously-generated poses might help combat regression to the mean, and has been used in motion generation without control inputs [Bütepage et al. 2017; Fragkiadaki et al. 2015; Zhou et al. 2018]. Zhang et al. [2018] use a similar approach to generate controllable quadruped motion, letting autoregressive and control information modify network weights, and demonstrate successful generation of both cyclic motion (gait) and simple non-cyclic motion such as jumping.

For many types of motion, no information is readily available that successfully disambiguates motion predictions. One example is co-speech gestures like head and hand motion, where the motion is unstructured and aperiodic and the dependence on the control signal (speech acoustics or transcriptions) is weak and nonlinear. The absence of strongly predictive input information means that deterministic motion-generation methods such as Ding et al. [2015]; Hasegawa et al. [2018]; Kucherenko et al. [2019]; Yoon et al. [2019] largely fail to produce distinct and lifelike motion.

2.3 Probabilistic data-driven motion synthesis

Probabilistic models represent another path to avoid collapsing on a mean pose: By building models of all plausible pose sequences given the available information (prior poses and/or control inputs), any randomly-sampled output sequence should represent convincing motion. As discussed in Sec. 2.1, many older models assume a Gaussian or Gaussian mixture distribution for poses given the state of the process, for example the (hidden) LSTM state. Conditional restricted Boltzmann machines (cRBMs) [Taylor and Hinton 2009; Taylor et al. 2011] are one example of this. The hidden state can also be made probabilistic. Examples include the SHMMs used for motion generation in Brand and Hertzmann [2000], locally linear models like switching linear dynamic systems (SLDSs) [Bregler 1997; Murphy 1998], Gaussian processes latent-variable models (GP-LVMs) [Lawrence 2005], and VAEs [Kingma and Welling 2014; Rezende et al. 2014]. Locally linear models were used for for motion synthesis in Chai and Hodgins [2005]; Pavlović et al. [2000], but have primarily been applied in recognition tasks. GP-LVMs and the closely related Gaussian process dynamical models (GPDMs) have been extensively studied in motion generation [Grochow et al. 2004; Levine et al. 2012; Wang et al. 2008] but they – along with other kernel-based motion-generation methods such as the radial basis functions (RBFs) in Kovar and Gleicher [2004]; Mukai and Kuriyama [2005]; Rose et al. [1998] – are unattractive in the big-data era since their memory and computation demands scale quadratically (or worse) in the number of training examples. VAEs circumvent computational issues by using a variational and amortised (see Cremer et al. [2018]) approximation of the likelihood for training. They have been applied to model controllable human locomotion [Habibie et al. 2017; Ling et al. 2020] and to generate head motion from speech [Greenwood et al. 2017a,b]. Ling et al. [2020] describes an autogregressive unconditional motion model based on VAEs, using a deterministic decoder based on the mixture-of-experts architecture from Zhang et al. [2018]. β-VAEs [Higgins et al. 2016] are used to mitigate posterior collapse, while scheduled sampling [Bengio et al. 2015] is necessary to stabilise long-term motion generation. Reinforcement learning is used to enable character control, although response time is somewhat sluggish. Notably, many VAE methods either generate noisy motion samples (e.g., Taylor et al. [2011]) or choose to not sample from the (Gaussian) observation distribution given the latent state of the process, instead generating the mean of the conditional Gaussian only [Greenwood et al. 2017a,b; Ling et al. 2020]. This risks re-introducing mean collapse and artificially reduces output entropy. We take this as evidence that these methods failed to learn an accurate and convincing motion distribution.

Variations of GANs [Sadoughi and Busso 2018] and adversarial training [Ferstl et al. 2019; Starke et al. 2020; Wang et al. 2019] have also been applied for motion generation and the related task of generating speech-driven video of talking faces [Pham et al. 2018; Pumarola et al. 2018; Vougioukas et al. 2018, 2020]. In contrast to GANs and VAEs, Starke et al. [2020] add latent-space noise to motion only at synthesis time (not during training), to obtain more varied motion, albeit at the expense of deviating from the desired input control. This approach also means that the distribution of the motion is not learned, and need not match that of natural motion.

Unlike previously-cited probabilistic motion-generation methods, GANs do not assume that observations are Gaussian given the state of the data-generating process. This avoids both regression towards the mean and Gaussian noise in output samples. The same goes for the discretisation-based approach in Sadoughi and Busso [2019], which learns a probabilistic model that triggers motion sequences from a fixed motion library. We consider another method for avoiding Gaussian assumptions, by introducing the first probabilistic motion model based on normalising flows. In contrast to MVAEs [Ling et al. 2020], our method can model conditional motion distributions, and so has controllability built in.

3 METHOD

This section introduces our new probabilistic motion model. The basic idea is to treat motion as a series of poses, and model these poses using an autoregressive model. In other words, we describe the conditional probability distribution of the next pose in the sequence as a function of previous poses and relevant control inputs. Like in a conditional GAN, the next pose of the motion is generated by drawing a random sample from a simple distribution such as a

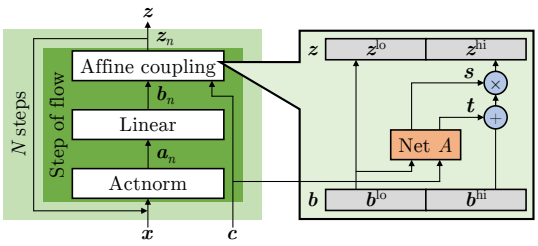

Fig. 2. Glow steps f_n^{-1} during inference. Detail of coupling layer on right.

Gaussian, and then nonlinearly transforming that sample by passing it through a neural network. This has the effect of reshaping the simple starting distribution into a more complex distribution that fits the distribution of the next pose in data. However, unlike a GAN, the neural network we use is invertible, which allows us to directly compute and maximise the likelihood of the data under the model. This makes the model stable to train. We now introduce basic notation and (in Sec. 3.1) describe how to construct normalising flows. Secs. 3.2 and 3.3 then detail, step by step, how to build a controllable autoregressive sequence model out of such flows.

For notation, we write vectors, and sequences thereof, in bold font. Upper case is used for random variables and matrices, and lower case for deterministic quantities or specific outcomes of the random variables. In particular, $\boldsymbol{X}$ typically represents randomly-distributed motion with $\boldsymbol{x} \in \mathsf{R}^{D\times T}$ being an outcome of the same, while $\boldsymbol{c} \in \mathsf{R}^{C\times T}$ represents the matching *control-signal inputs*, which in our experiments are relative and rotational velocities that describe motion along path on the ground plane. Non-bold capital letters generally denote indexing ranges, with matching lower-case letters representing the indices themselves, e.g., $t \in \{1, \ldots, T\}$. Indices into sequences extract specific time frames, for example individual *poses* $\boldsymbol{x}_t \in \mathsf{R}^D$, or sub-sequences $\boldsymbol{x}_{1:t} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t]$. Each pose parameterises the positions and orientations of objects such as a whole body, parts of a body, or keypoints on a body or face. In this paper, the pose vector $\boldsymbol{x}_t$ is created by concatenating vectors that represent either joint positions or joint rotations on a 3D skeleton.

3.1 Normalising flows and Glow

Normalising flows are flexible generative models that allow both efficient sampling and efficient inference. The idea is to subject samples from a simple, fixed *base* (or *latent*) distribution Z on R^D to an invertible and differentiable nonlinear transformation $\boldsymbol{f} : \mathsf{R}^D \to \mathsf{R}^D$, in order to produce samples from a new, more complex distribution $\boldsymbol{X}$. If this transformation has many degrees of freedom, a wide variety of different distributions can be described.

Flows construct expressive transformations $\boldsymbol{f}$ by chaining together numerous simpler nonlinear transformations $\{\boldsymbol{f}_n\}_{n=1}^N$, each of them parameterised by a $\boldsymbol{\theta}_n$ such that $\boldsymbol{\theta} = \{\boldsymbol{\theta}_n\}_{n=1}^N$. We define the observable random variable $\boldsymbol{X}$, the latent random variable $\boldsymbol{Z} \sim \mathcal{N}(\mathbb{0}, \boldsymbol{I})$, and intermediate distributions $\boldsymbol{Z}_n$ as follows:

$$\boldsymbol{z} = \boldsymbol{z}_N \overset{\boldsymbol{f}_N}{\to} \boldsymbol{z}_{N-1} \overset{\boldsymbol{f}_{N-1}}{\to} \ldots \overset{\boldsymbol{f}_2}{\to} \boldsymbol{z}_1 \overset{\boldsymbol{f}_1}{\to} \boldsymbol{z}_0 = \boldsymbol{x} \tag{1}$$

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{f}_1 \ \boldsymbol{f}_2 \ \ldots \boldsymbol{f}_N(\boldsymbol{z}) \tag{2}$$

$$\boldsymbol{z}_n(\boldsymbol{x}) = \boldsymbol{f}_n^{-1} \circ \ldots \circ \boldsymbol{f}_1^{-1}(\boldsymbol{x}). \tag{3}$$

The sequence of (inverse) transformations $\boldsymbol{f}_n^{-1}$ in (3) is known as a *normalising flow*, since it transforms the distribution $\boldsymbol{X}$ into an isotropic standard normal random variable $\boldsymbol{Z}$.

Similar to the generators in GANs, normalising flows are *implicit probabilistic models* according to the definition in Mohamed and Lakshminarayanan [2016]. While explicit models draw samples from probability density functions defined in the space of the observations, GANs and normalising flows instead generate output by drawing samples $\boldsymbol{z}$ from a latent base distribution $\boldsymbol{Z}$ that acts as a source of entropy, and then subjecting these samples to a deterministic, nonlinear transformation $\boldsymbol{f}$ to obtain samples $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{z})$ from $\boldsymbol{X}$. Unlike GANs, however, normalising flows permit fast and easy probability computation (inference), since the transformation $\boldsymbol{f}$ is invertible: Using the change-of-variables formula, we can write the log-likelihood of a sample $\boldsymbol{x}$, as used in likelihood maximisation, as

$$\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \ln p_{\mathcal{N}}\ \boldsymbol{z}_N(\boldsymbol{x}) + \sum_{n=1}^{N} \ln \det \frac{\partial \boldsymbol{z}_n(\boldsymbol{x})}{\partial \boldsymbol{z}_{n-1}}, \tag{4}$$

where $\frac{\partial \boldsymbol{z}_n(\boldsymbol{x})}{\partial \boldsymbol{z}_{n-1}}$ is the Jacobian matrix of $\boldsymbol{f}_n^{-1}$ at $\boldsymbol{x}$, which depends on $\boldsymbol{\theta}$, and $p_{\mathcal{N}}$ is the probability density function of the D-dimensional standard normal distribution. The general determinant in (4) has computational complexity close to $O(D^3)$, so many improvements to normalising flows involve the development of $\boldsymbol{f}_n$-transformations with tractable Jacobian determinants, that nonetheless yield highly flexible transformations under iterated composition. An in-depth review of normalising flows and different flow architectures can be found in Papamakarios et al. [2019]. In this work, we consider the *Glow* architecture [Kingma and Dhariwal 2018], first developed for images, and extend it to model controllable motion sequences.

Each component transformation $\boldsymbol{f}_n^{-1}$ in Glow contains three sub-steps: *activation normalisation*, also known as *actnorm*; a *linear transformation*; and a so-called *affine coupling layer*, together shown as a *step* of flow in in Fig. 2. The first two are affine or linear transformations while the latter amounts to a more powerful nonlinear transformation that is nonetheless invertible.

We will let $\boldsymbol{a}_{t,n}$ and $\boldsymbol{b}_{t,n}$ denote intermediate results of Glow computations for observation $\boldsymbol{x}_t$ in flow step n, as shown in Fig. 2. Actnorm, the first sub-step, is an affine transformation $\boldsymbol{a}_{t,n} = \boldsymbol{s}_n \odot \boldsymbol{z}_{t,n-1} + \boldsymbol{t}_n$ (with $\odot$ denoting elementwise multiplication) intended as a substitute for batchnorm [Ioffe and Szegedy 2015]. The parameters $\boldsymbol{s}_n \neq 0$ and $\boldsymbol{t}_n$ are initialised such that the output has zero mean and unit variance and then treated as trainable parameters. After actnorm follows a linear transformation $\boldsymbol{b}_{t,n} = \boldsymbol{W}_n \boldsymbol{a}_{t,n}$ where $\boldsymbol{W} \in \mathsf{R}^{D\times D}$. By representing $\boldsymbol{W}_n$ by an LU-decomposition $\boldsymbol{W}_n = \boldsymbol{L}_n \boldsymbol{U}_n$ with one matrix diagonal set to one (say $l_{n,dd} = 1$), the Jacobian determinant of the sub-step is just the product of the diagonal elements $u_{n,dd}$, which is computable in linear time. The non-fixed elements of $\boldsymbol{L}_n$ and $\boldsymbol{U}_n$ are the trainable parameters of the sub-step.

The affine coupling layer is more complex. The idea is to affinely transform half of the input elements based on the values of the other half. By passing those remaining elements through unchanged, it is easy to use their values to undo the transformation when reversing the computation. Mathematically, we define $\boldsymbol{b}_{t,n}$ and $\boldsymbol{z}_{t,n}$ as concatenations $\boldsymbol{b}_{t,n} = [\boldsymbol{b}_{t,n}^{\text{lo}}, \boldsymbol{b}_{t,n}^{\text{hi}}]$ and $\boldsymbol{z}_{t,n} = [\boldsymbol{z}_{t,n}^{\text{lo}}, \boldsymbol{z}_{t,n}^{\text{hi}}]$. The coupling

can then be written

$$[z^{\mathrm{lo}}_{t,n},\, z^{\mathrm{hi}}_{t,n}] = [\boldsymbol{b}^{\mathrm{lo}}_{t,n},\, (\boldsymbol{b}^{\mathrm{hi}}_{t,n} + \boldsymbol{t}'_{t,n}) \odot s'_{t,n}]. \tag{5}$$

The scaling $s'_n > 0$ and bias $\boldsymbol{t}'_n$ terms in the affine transformation of the $\boldsymbol{b}^{\mathrm{hi}}_{t,n}$ are computed via a neural network, A_n, that only takes $\boldsymbol{b}^{\mathrm{lo}}_{t,n}$ as input. (We use 'A' for "affine".) We can therefore unambiguously invert Eq. (5) based on $\boldsymbol{z}_{t,n}$ by feeding $\boldsymbol{z}^{\mathrm{lo}}_{t,n} = \boldsymbol{b}^{\mathrm{lo}}_{t,n}$ into A_n to compute $s'_n > 0$ and $\boldsymbol{t}'_n$. The coupling computations during inference are visualised in Fig. 2. The weights that define A_n are also elements of the parameter set $\boldsymbol{\theta}_n$, while the constraint $s'_n > 0$ is enforced by using a sigmoid nonlinearity [Nalisnick et al. 2019, App. D]. Random weights are used for initialisation except in the output layer, which is initialised to zero [Kingma and Dhariwal 2018]; this has the effect that the coupling initially is close to an identity transformation, reminiscent of Fixup initialisation [Zhang et al. 2019].

Interleaved linear transformations and couplings are both necessary for an expressive flow. Without couplings, a stack of flows collapses to compute a single, fixed affine transformation of $\boldsymbol{Z}$, meaning that $\boldsymbol{X}$ will be restricted to a Gaussian distribution; a stack of couplings alone will only perform a nonlinear transformation of *half* of $\boldsymbol{Z}$, doing nothing to the other half. The linear layers $\boldsymbol{W}_n$ can be seen as generalised permutation operations between couplings, ensuring that all variables (not just one half) can be nonlinearly transformed with respect to each other by the full flow.

3.2 MoGlow

Let $\boldsymbol{X} = \boldsymbol{X}_{1:T} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_T]$ be a sequence-valued random variable. Like all autoregressive models of time sequences, we develop our model from the decomposition

$$p(\boldsymbol{x}) = p(\boldsymbol{x}_{1:\tau}) \prod_{t=\tau+1}^{T} p(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}). \tag{6}$$

We assume the distribution $\boldsymbol{X}_t$ only depends on the τ previous values (i.e., is a Markov chain of order τ), except for a latent state $\boldsymbol{h}_t \in \mathsf{R}^H$ that represents the effect of recurrence in a recurrent neural network (RNN) and evolves according to a relation $\boldsymbol{h}_t = \boldsymbol{g}(\boldsymbol{h}_{t-1})$ at each timestep. To achieve control over the output we further condition the $\boldsymbol{X}$-distribution on another sequence variable $\boldsymbol{C}$, acting as the *control signal*. We assume that, for each training-data frame $\boldsymbol{x}_t$, the matching control-signal values $\boldsymbol{c}_t \in \mathsf{R}^C$ are known. Moreover, the experiments in this paper focus on causal control schemes, where only current and former control inputs $\boldsymbol{c}_{1:t}$ may influence the conditional distributions from (6) at t. (Letting the model also depend on future $\boldsymbol{c}$-values might improve motion quality, but inevitably introduces algorithmic latency.) Putting the Markov assumption, the hidden state, and the control together gives our temporal model

$$p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{c}) = p(\boldsymbol{x}_{1:\tau} \mid \boldsymbol{c}_{1:\tau}) \prod_{t=\tau+1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-\tau:t-1}, \boldsymbol{c}_{t-\tau:t}, \boldsymbol{h}_{t-1}) \tag{7}$$

$$\boldsymbol{h}_t = \boldsymbol{g}_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-\tau:t-1}, \boldsymbol{c}_{t-\tau:t}, \boldsymbol{h}_{t-1}), \tag{8}$$

where we have decided to condition on the control signal at most τ steps back only, just like for the previous poses. The subscript $p_{\boldsymbol{\theta}}$ indicates that the distributions depend on model parameters $\boldsymbol{\theta}$. The initial hidden state can be learned, but in our experiments we initialise $\boldsymbol{h}_\tau$ as 0.[1] For the deterministic hidden-state evolution $\boldsymbol{g}$ a straightforward choice to implement Eq. (8) is to use a recurrent neural network, here an LSTM [Hochreiter and Schmidhuber 1997]. The vector $\boldsymbol{h}_t$ is then the concatenation of the LSTM cell state vectors and the LSTM-unit output vectors at time t.

Finally, we also assume *stationarity*, meaning that $\boldsymbol{g}$ and the distributions in (7) are independent of t. This is an exceedingly common assumption in practical sequence models, since it means that all timesteps in the training data can be treated as samples from a single, time-independent distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-\tau:t-1}, \boldsymbol{c}_{t-\tau:t}, \boldsymbol{h}_{t-1})$. The central innovation in this paper is to learn that controllable *next-step distribution* using normalising flows.

To adapt Glow to parameterise the next-step distribution in the autoregressive hidden-state model in Eqs. (7) and (8), we made a number of changes to the original image-oriented Glow architecture in Kingma and Dhariwal [2018]. There, dependencies between $\boldsymbol{Z}_{t,n}$-values at different image locations were introduced by making A_n a convolutional neural network. We instead use unidirectional (causal) LSTMs inside A_n to enable dependence between timesteps, which is simpler than the dilated convolutions used in recent audio models based on Glow [Kim et al. 2019; Prenger et al. 2019] while giving better models than making A_n a simple feedforward network.

We added a small epsilon $\varepsilon = 0.05$ to the sigmoids in A_n that define the scale-factor outputs s'_n, in order to bound the dynamic range of the scaling and stabilise training. This modification restricts the possible scale-factor values to the interval $s_n \in (\varepsilon, 1+\varepsilon)$. Unlike Dinh et al. [2017]; Kim et al. [2019] we did not use any multiresolution architecture in our flow, as that did not provide any noticeable improvements in preliminary experiments, nor do we include squeeze operations, as that would add algorithmic latency.

To provide motion control and enable explicit dependence on recent pose history in Glow distributions, we take inspiration from recent sequence-to-sequence audio models Kim et al. [2019]; Prenger et al. [2019], which feed the conditioning information (here $\boldsymbol{x}_{t-\tau:t-1}$ and $\boldsymbol{c}_{t-\tau:t}$) as additional inputs to the affine couplings A_n, these being the only neural networks in Glow. The scaling and bias terms, together with the next state $h_{t,n}$ of net A_n, are then computed as

$$[s'_{t,n},\, \boldsymbol{t}'_{t,n},\, \boldsymbol{h}_{t,n}] = A_n(\boldsymbol{b}^{\mathrm{lo}}_{t,n},\, \boldsymbol{x}_{t-\tau:t-1},\, \boldsymbol{c}_{t-\tau:t},\, \boldsymbol{h}_{t-1,n}). \tag{9}$$

We call our proposed model structure *MoGlow* for *motion Glow*.

If we let $\boldsymbol{z}_{t,N}$ denote the observation $\boldsymbol{x}_t$ mapped back onto the latent space by the (conditional) flow transformation $\boldsymbol{f}^{-1}$, the full log-likelihood training objective of MoGlow applied to a sequence $\boldsymbol{x}$ given the control input $\boldsymbol{c}$ can be written

$$\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}_{\tau+1:T} \mid \boldsymbol{x}_{1:\tau}, \boldsymbol{c}) = \sum_{t=\tau+1}^{T} \ln p_{\mathcal{N}}\; \boldsymbol{z}_{t,N}(\boldsymbol{x}_{1:t}, \boldsymbol{c}_{1:t}) + \sum_{n=1}^{N} \sum_{d=1}^{D} \sum_{t=\tau+1}^{T} \ln s_{n,d} + \ln u_{n,dd} + \ln s'_{t,n,d}(\boldsymbol{x}_{1:t},\, \boldsymbol{c}_{1:t}), \tag{10}$$

where we have made explicit which terms depend on $\boldsymbol{x}$ and $\boldsymbol{c}$. A schematic illustration of MoGlow sample generation is presented in

[1] For this article, we will ignore how to model the initial distribution $p(\boldsymbol{x}_{1:\tau})$ from (7). Experimentally, we found that initialisation with natural motion snippets or with a static mean pose both give competitive results.

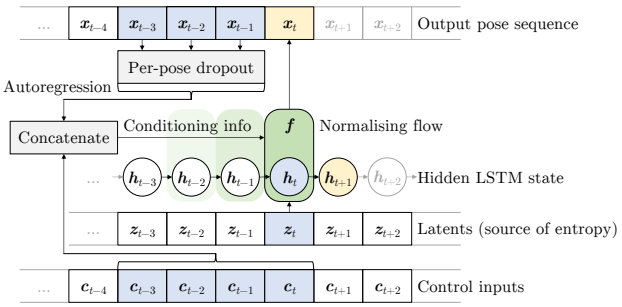

Fig. 3. Schematic of autoregressive motion generation with MoGlow. Inputs are blue, outputs yellow. Dropout is only applied at training time.

Fig. 3. At generation time, latent z_t-vectors are sampled independently from $p_{\mathcal{N}}$ (acting as a source of randomness for the next-step distribution) and then transformed into new poses $\boldsymbol{x}_t$ by the flow $\boldsymbol{f}$ conditioned on $\boldsymbol{x}_{t-\tau:t-1}$, $\boldsymbol{c}_{t-\tau:t-1}$, and $\boldsymbol{h}_{t-1}$.

Because $\boldsymbol{Z}_t$ is supported on all of R^D, so is $\boldsymbol{X}_t$. This is a natural fit for pose representations that take values on R^D, e.g., joint positions in Cartesian coordinates. Pose representations supported on a non-zero volume subset $\mathcal{X} \subset \mathsf{R}^D$, for example the exponential map [Grassia 1998], can also be used. In practise, we recommend parameterisations that minimise angular discontinuities, e.g., by expressing angles relative to a T-pose and wrapping at ±180 degrees, since the method works best for continuous density functions.

3.3 Data dropout

Early MoGlow models had a problem with poor adherence to the control input, where generated character motion often would walk or run even when the control input (in this case, the path followed by the root node) specified that no movement through space was taking place. This indicates an over-reliance on autoregressive pose information, compared to the control input. Such behaviour is a frequent issue with long-term prediction in powerful autoregressive models (cf. Chen et al. [2017]; Liu et al. [2019]), for example in generative models of speech as in Tachibana et al. [2018]; Uria et al. [2015]; Wang et al. [2018]. Established methods to counter this failure mode include applying dropout to entire frames of autoregressive history inputs – conventionally called *data dropout* – as in Bowman et al. [2016]; Wang et al. [2018], or downsampling the data sequences as in Tachibana et al. [2018]. Dropout and bottlenecks in the autoregressive path can also be combined with a lowered frame rate, e.g., Shen et al. [2018]; Wang et al. [2017]. All of these approaches have the net effect of reducing the informational value of the most-recent autoregressive feedback, thus making the information in the current control input relatively more valuable. We found that applying data dropout during training substantially improved the consistency between the generated motion and the control signal in MoGlow models. In particular, the issue of MoGlow running in place vanished with frame dropout rates of 50% and above.

4 EXPERIMENTAL SETUP

The goal of MoGlow is to introduce a probabilistic and controllable motion model capable of delivering high-quality output without task-specific assumptions. This section presents data and systems used for comparative experiments that evaluate the quality of MoGlow output across different tasks. Associated evaluations and results are reported in Sec. 5, along with skinned-character experiments designed to validate the probabilistic aspects of the model.

Objectively evaluating motion plausibility is difficult in the general case, as there is no single natural realisation of the motion given typical, weak control signals. Comparing low-level properties such as frame-wise joint positions between recorded and synthesised motion is therefore not particularly informative. To enable meaningful objective evaluation, we chose to evaluate MoGlow on locomotion synthesis, for which some perceptually-salient aspects of the motion can be studied objectively. Specifically, foot-ground contacts are easy to identify as they should have zero velocity, and foot-sliding artefacts (often attributable to mean collapse) are both pervasive in synthetic locomotion and known to greatly affect the perceived naturalness of the resulting animation. We stress that unlike Holden et al. [2017, 2016]; Pavllo et al. [2018]; Starke et al. [2020], we do not use foot-contact information as part of our model, but only use it to objectively evaluate the generated output motion.

4.1 Data for objective and subjective evaluations

We considered two sources of motion-capture data in our evaluations, namely human (bipedal) and animal (quadrupedal) locomotion on flat surfaces. Bipedal and quadrupedal locomotion represent significantly different modelling problems, and to our knowledge no method has been demonstrated to perform well on both tasks, with the exception of Starke et al. [2020], which appeared while this paper was in review. For the human data, we used the data and preprocessing code provided by Holden et al. [2016, 2015].[2] We pooled this dataset with the locomotion trials from the CMU [CMU Graphics Lab 2003] and HDM05 [Müller et al. 2007] databases. We held out a subset of the data with a roughly equal amount of motions in different categories (such as walking, running, and sidestepping) for evaluation, and used the rest for training. For the animal motion, we used the 30 minutes of dog motion capture from Zhang et al. [2018], excluding clips on uneven terrain. Quadrupedal locomotion allows more gaits than bipedal locomotion (see Zhang et al. [2018]), but the data also contains motions like sitting, standing, idling, lying, and jumping. We held out two sequences comprising 72 s of data.

Both datasets were downsampled to 20 frames per second and sliced into fixed-length 4-second windows with 50% overlap for training. The lowered frame rate both reduces computational demands and decreases over-reliance on autoregressive feedback, as discussed in Sec. 3.3. The training data was subsequently augmented by lateral mirroring. To increase the amount of backwards and side-stepping motion, we further augmented the data by reversing it in time. This way we obtained 13,710 training sequences from the human data and 3,800 from the animal material. Preliminary comparisons indicated that the reverse-time augmentation substantially improved the naturalness of synthesised motion.

We used the same pose representation and control scheme as in Habibie et al. [2017]. Each pose frame $\boldsymbol{x}_t$ in the data thus comprised 3D joint positions of a skeleton expressed in a floor-level (root)

[2]Please see http://theorangeduck.com/page/deep-learning-framework-character-motion-synthesis-and-editing.

Table 1. Overview of system configurations considered in this paper. Numbers with [h] pertain only to the human model, [d] to the dog.

	Configuration	ID	Proba-bilistic?	Task-agnostic?	Algo. latency	Context frames	Hidden state	Pose dropout	Num. params. Man	Num. params. Dog	Training… Loss func.	Epochs	Time	GPUs
Baselines	Plain LSTM	RNN	✗	✓	None	-	LSTM	-	1M	1M	MSE	40	0.7[h]	8
	Greenwood et al. [2017a]	VAE	Partially	✓	Full seq.	-	BLSTM	-	4M	4M	MSE+KLD	40	6.1[h]	8
	Pavllo et al. [2018]	QN	✗	✗	1 sec.	-	GRU	-	10M	-	Angl./pos.+reg.	2k/4k	10[h]	2
	Zhang et al. [2018]	MA	✗	✗	1 sec.	12	-	-	-	5M	MSE	150	30[d] h	1
	MoGlow	MG	✓	✓	None	10	LSTM	95%	74M	80M	Log-likelihood	291[h]	26[h]	1
Ablats.	No pose dropout	MG-D	"	"	"	10	"	0%	74M	-	"	"	26[h]	"
	No pose context	MG-A	"	"	"	10	"	100%	74M	-	"	"	26[h]	"
	Minimal history	MG-H	"	"	"	1	"	95%	54M	-	"	"	23[h]	"

coordinate system following the character's position and direction. The root motion was calculated by Gaussian-filtering the horizontal, floor-projected hip motion from the original data, which yielded a (x, z) trajectory on the ground together with the up (y) axis rotation. The filtering is essential for generalising the synthesis to smooth control signals as provided by an artist or from game-pad input.

The human data had 21 joints ($D = 63$ degrees of freedom), while the dog data had 27 joints ($D = 81$ degrees of freedom). This was supplemented with the frame-wise delta translation and delta rotation (around the up-axis) of the root, which together constitute the control signal $\boldsymbol{c}_t \in \mathsf{R}^3$ for each frame. The trajectory of the root over time is computed from the control signal $\boldsymbol{c}_t$ using integration, and is therefore completely determined by the sequence of control inputs $\boldsymbol{c}$. The end result is that the root is constrained to exactly follow a specific path on the ground and path-following is essentially perfect; the task of the motion-synthesis model is to generate a sequence of body poses that are consistent with motion along this trajectory. Each dimension in the data and control signal was standardised to zero mean and unit variance over the training data prior to training.

4.2 Proposed model and ablations

We trained the same PyTorch implementation[3] of MoGlow on both the human and the animal data. We used a $\tau = 10$-frame time window (0.5 seconds) with $N = 16$ steps of flow. The neural network in each coupling layer comprised two LSTM layers (512 nodes each), followed by a linear (for $\boldsymbol{t}_n$) and sigmoid (for $\boldsymbol{s}_n$) output layer. Model parameters were estimated by maximising the log-likelihood of the training-data sequences using Adam [Kingma and Ba 2015] for 160k steps (human) or 80k (quadruped) with batch size 100. Both models used a learning rate of 10^{-4}, but for the quadruped we used the Noam learning rate scheduler [Vaswani et al. 2017] with 1k steps of warm-up and peak learning rate 10^{-3}. The autoregressive frame dropout rate was set to 0.95 during training (no dropout was used during synthesis). We denote this system "MG" for MoGlow. While many GANs and normalising-flow applications heuristically reduce the temperature (standard deviation) of the latent distribution $\boldsymbol{Z}_t$ at generation time, we found this to be unnecessary, and in fact detrimental to the visual quality of motion sampled from the system.

[3]Please see our project page https://simonalexanderson.github.io/MoGlow for links to code, data, and hyperparameters from the evaluation, as well as updated hyperparameter settings that we think further improve output quality.

To assess the impact of important design decisions, we trained three additional versions of the MoGlow architecture on the human data. In these, specific components had been disabled from the full MG system: The first ablated configuration, "MG-D" (for "minus dropout") turned off data dropout by setting the dropout rate to zero. As discussed in Sec. 3.3, we expect this system to exhibit poor adherence to the control signal and establish the utility of introducing data dropout. The second, "MG-A" (for "minus autoregression"), instead increased the dropout rate to 100%, thereby completely disabling autoregressive feedback from recent poses $\boldsymbol{x}_{t-\tau:t-1}$. We expect the contrasts between MG and MG-A to show the utility of the autoregressive feedback in the model. The final ablation, "MG-H" (for "minus history") changed τ from ten frames (0.5 s of history information) down to a single frame. This is the minimum history length at which the model remains autoregressive; any pose or control information older than $t - 1$ must now be propagated by the LSTMs in A_n instead. (Unlike MG-D and MG-A, MG-H also affects the control information, in addition to the autoregressive feedback.) We expect this ablation to demonstrate the utility of providing the flows with an explicit memory buffer of the most recent pose and control inputs, in addition to the long-range information about past inputs propagated through the recurrent hidden state. Table 1 summarises the properties and training of the proposed system and its ablations.

4.3 Baseline systems

To put the performance of MoGlow in perspective, we compared against a number of other motion generation approaches. The first of these is held-out motion capture recordings, which we label "NAT" for natural. (We prefer not to use the term "ground truth", since there is no one true way to perform a given motion.) These motion examples function as a top line.

We also compared against two task-agnostic motion-synthesis approaches, labelled "RNN" and "VAE". The first of these, RNN, is a deterministic system that maps control signals $\boldsymbol{c}_t$ to poses $\boldsymbol{x}_t$ using a standard unidirectional LSTM network (one hidden layer of 512 nodes followed by a linear output layer) and was trained to minimise the mean squared error (MSE). Because our path-based control signal does not suffice to disambiguate the motion, we expect this generic method to exhibit considerable regression to the mean, for instance visible through foot-sliding. This is emblematic of task-agnostic deterministic methods. The other task-agnostic baseline,

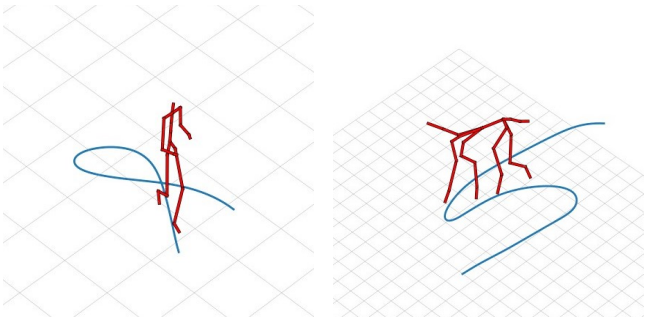

Fig. 4. Still images cropped from videos of MG output. The path followed by the root node, which is completely determined by $\boldsymbol{c}$, is visualised as a blue curve projected onto the ground plane.

VAE, is a reimplementation of the conditional variational autoencoder architecture used for speech-driven head-motion generation in Greenwood et al. [2017a,b], but in our case predicting motion $\boldsymbol{x}$ from $\boldsymbol{c}$. We used encoders and decoders with two bidirectional LSTM layers (256 nodes each way) and a linear output layer. The encoder used mean-pooling to map to a latent space with two dimensions per sequence. Due to the bidirectional LSTMs in the conditional decoder, interactive control is not possible with this approach. Unlike the RNN baseline, VAE represents a partially probabilistic model, which should enable it to cope with motion that is random and ambiguous also when conditioned on the control signal. The model does not incorporate any assumptions specific to head-motion data, and can be considered representative of the state of the art in probabilistic, task-agnostic motion generation. We say that this system is "partially probabilistic" since the decoder is trained to minimise the MSE and treated as deterministic rather than stochastic at synthesis time. As a consequence, output samples from the system have artificially reduced randomness compared to sampling from the full probabilistic model described by the fitted VAE, whose decoder is a Gaussian distribution. Such reduced-entropy generation procedures are common in practice since they tend to improve subjective output quality (see Sec. 2.3), but also indicate that the underlying model has failed to convincingly model the natural variation in the data.

Finally, we also compared our proposed method with a leading task-specific system in each of the two domains. Human locomotion generation, to begin with, is a mature field where many approaches may be considered state-of-the-art. One example is the recently proposed QuaterNet [Pavllo et al. 2018], which we included in our evaluation as system "QN". In order not to compromise QN motion quality, we used the code, hyperparameters, and control scheme made available by the original QuaterNet authors.[4] This introduced a number of minor differences compared to other systems. Specifically, the QuaterNet reference implementation contains a number of preprocessing steps that change the motion: First the input path is approximated by a spline, and facing information and local motion speed are replaced. This control scheme causes the character to always face the direction of motion, preventing sidestepping or walking backwards. Short spline segments are then lengthened, preventing the model from standing still. One goal with MoGlow is to deliver high-quality motion without such custom, task-specific processing steps. Finally, we resampled the output from the trained QN system to 20 fps to match the other systems in the evaluation.

For the quadruped locomotion task, we compared with the mode-adaptive neural networks from Zhang et al. [2018]. Since they trained on the same dataset as us, we used their pretrained model[5] as our system "MA" for best results. To our knowledge, no data was held out from their training. In the absence of held-out control signals, MA was therefore only evaluated on synthetic control input. For the experiments we set the MA style input to "move" and the correction parameter τ to 1, to make the model follow the input patch exactly, like the other systems in the evaluation. MA output was also resampled to 20 fps.

In summary, RNN and VAE are task-agnostic systems – one deterministic, one probabilistic – while QN and MA instead represent the task-specific state of the art in their respective task. We note that, unlike RNN and the MG systems, VAE, QN, and MA are noncausal, in the sense that their output depends on future control-input information. We expect this ability to "see the future" to benefit the quality of the motion generated for these systems, but it comes at the cost of introducing algorithmic latency, preventing the type of responsive control that MG allows. All our models were trained on a system with 8 Nvidia 2080Ti GPUs. An overview of the different systems, including information such as training time, model size, and the number of GPUs used, is provided in Table 1.

5 RESULTS AND DISCUSSION

This section details our subjective (Secs. 5.1 through 5.2) and objective (Sec. 5.3) evaluations of the different motion-generation methods from Sec. 4, and how we interpret the results. We then describe (Sec. 5.4) experiments that explore the probabilistic aspects of the model, and consider its use beyond locomotion. We then conclude with a discussion of drawbacks and limitations (Sec. 5.5).

5.1 Subjective evaluation setup

Since our goal is to create lifelike synthetic motion that appears convincing to human observers, subjective evaluation is the gold standard. To this end we conducted several user studies to measure motion quality on the two tasks. The stimuli used in both studies were short animation clips where motion was visualised using a stick figure seen from a fixed camera angle; see Fig. 4. A curve on the ground marked the path taken by the figure in the clip. Clips were generated for all systems in Table 1 and from held-out motion-capture recordings ("NAT"). For MG, one second of preceding motion was pre-generated before the four seconds that were displayed and scored, to remove the effects of motion initialisation. Since the QuaterNet preprocessing changes the motion duration, the segmentation points for the evaluation clips (and also the camera azimuth) differ between QN and the other systems.

In addition to motion generated from held-out natural control signals (20 human, 8 dog), the evaluation also included synthetic control signals (7 human, 10 dog) with a range of motion speeds and directions, for which no natural counterpart was available. Generalising well to synthetic control is important for computer animation, video games, and similar applications.

[4] Please see https://github.com/facebookresearch/QuaterNet.

[5] Available at https://github.com/sebastianstarke/AI4Animation.

Table 2. Mean subjective ratings with confidence intervals. Significant differences from MG are indicated by ** ($p < 0.01$) and * ($p < 0.05$).

ID	Human		Quadruped	
	Held-out $\boldsymbol{c}$	Synthetic $\boldsymbol{c}$	Held-out $\boldsymbol{c}$	Synthetic $\boldsymbol{c}$
NAT	4.27±0.11	-	4.25 ± 0.06**	-
RNN	3.10±0.15**	1.9±0.2**	2.81 ± 0.10**	1.14 ± 0.04**
VAE	3.95±0.13	3.1±0.3**	3.55 ± 0.08	2.14 ± 0.20**
QN	4.21±0.10	-	-	-
MA	-	-	-	3.78 ± 0.10
MG	4.17±0.11	4.0±0.2	3.71 ± 0.18	3.57 ± 0.20
MG-D	3.66±0.16**	2.1±0.2**	-	-
MG-A	2.86±0.16**	3.2±0.3**	-	-
MG-H	3.87±0.13*	3.9±0.3	-	-

Evaluation participants were recruited using the *Figure Eight* crowdworker platform at the highest-quality contributor setting (allowing only the most experienced, highest-accuracy contributors). For each clip, participants were asked to grade the perceived naturalness of the animation on a scale of integers from 1 to 5, with 1 being *completely unnatural* (motion could not possibly be produced by a real person/dog) and 5 being *completely natural* (looks like the motion of a real person/dog). Every system in Table 1 had one stimulus generated for every control signal considered, with a few exceptions: QN was not applied to synthetic control signals, since these contained a large fraction of control inputs involving walking sideways, backwards, and standing still, motion that the QN reference implementation from Pavllo et al. [2018] cannot perform (instead rendering these as forwards motion). MA was not applied to our natural test inputs, since these were not held out from MA training. The ablated systems were only evaluated on the human locomotion task. This yielded a total of 202 human animations being evaluated (160 with held-out control and 42 with synthetic control) and 72 dog animations (32 held-out, 40 synthetic control). The order of the animation clips was randomised, and no information was given to the raters about which system had generated a given video, nor about the number of systems being evaluated in the test.

Interspersed among the regular stimuli were a handful of clips with deliberately *bad* animation taken from early iterations in the training process (labelled "BAD"). These were added as "attention checks" to be able to filter out unreliable raters: Any rater that had given any one of the BAD animations a rating of 4 or above, *or* had given any of the NAT clips a rating below 2, was removed from the analysis. Ratings that were too fast (the rater replied before the video had finished playing) were also discarded. Prior to the start of the rating phase, participants were trained by viewing example motion videos from the different conditions evaluated, as well as some of the bad examples mentioned above. Motion examples can be seen in our presentation video and in the supplementary material, which contains all video clips from the subjective evaluation.

5.2 Analysis and discussion of subjective evaluation

A total of 645 raters (296 human data/349 dog data) participated in the evaluation, of which 89 (49/40) were removed as unreliable (see above). In total, 10,355 ratings were collected (5,083/5,272). 1,533/983 of these were discarded due to unreliable rater (1,344/813) or too fast response time (189/170), resulting in a total of 3,550/4,289 ratings across 227/80 clips being evaluated (both regular and BAD), amounting to between 8 and 60 ratings per stimulus. The mean scores for each system configuration and control-signal class are tabulated in Table 2.

For the human motion, a one-way ANOVA revealed a main effect of the naturalness rating ($F = 223$, $p \ddot{Y} 10^{-288}$). A post-hoc Tukey multiple-comparisons test was applied in order to identify significant differences between conditions (FWER = 0.05). For the held-out control conditions, MG was rated significantly higher than RNN and all ablations. For the synthetic control conditions, MG was rated significantly higher than all other systems except the ablation system MG-H. The same analysis for the quadruped motion again revealed a main effect of the naturalness rating ($F = 172$, $p \ddot{Y} 10^{-100}$ for held-out $\boldsymbol{c}$, $F = 803$, $p \ddot{Y} 10^{-296}$ for synthetic). The post-hoc Tukey multiple-comparisons test revealed significant differences between MG and all other systems, except between MG and VAE on the held-out control and between MG and MA on the synthetic control. 95%-confidence intervals for the mean scores based on these analyses are included in Table 2, which also indicates significant differences between MG and other systems.

Among the task-agnostic methods in the experiment, MG substantially outperforms both RNN and VAE. Despite these MG systems being trained to predict joint positions rather than joint rotations, they are seen to respect constraints due to bone lengths, ground contacts, etc. Furthermore, the rated motion quality of MG on each task is comparable to the respective task-specific state of the art (the difference between MG and either QN or MA is not statistically significant), and comes within 0.1 points of natural motion for the biped. This is despite the task-specific systems having a full second of algorithmic latency, while MG is task-agnostic and has none. We note that stimuli where the root is completely still are generally rated lowest for MG and MA, and not possible to generate with QN.

Among other results, the performance of the ablations MG-D and MG-A versus the full MG system indicate that both autoregression and data dropout are of great importance for synthesising natural motion. A longer memory length of $\tau = 10$ frames for MG, compared to $\tau = 1$ for MG-H, also benefited the model. It can be observed that RNN, VAE, and MG-D quality degrades substantially on synthetic control signals, creating a highly significant difference with respect to MG. We hypothesise that this, for MG-D, is due to artefacts of poor control without data dropout (such as running in place; see Sec. 3.3), and, for RNN and VAE, due to the systems being dependent on footfall cues (e.g., residual periodicity in the root-node motion) not present in the synthetic motion control. The full MoGlow model, in contrast, generalises robustly to synthetic control signals.

5.3 Objective evaluation

Given the salience and importance of foot-sliding artefacts in locomotion synthesis, we base our objective evaluation on footstep analysis, with footsteps estimated as time intervals where the horizontal speed of the heel joints (bipeds) or toe joints (quadrupeds) are below a specified tolerance value v_{tol}. At low values of v_{tol}, many ground contacts exhibit too much motion (due to foot sliding

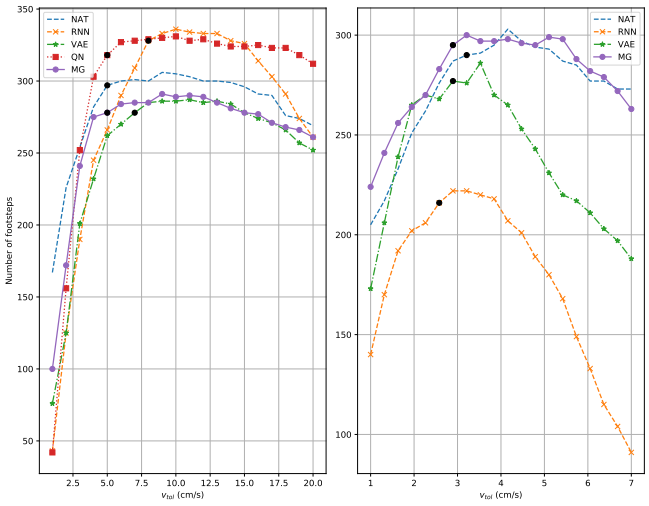

Fig. 5. Footstep count f_{est} as a function of speed tolerance v_{tol} (cm/s) for the human (left) and quadruped (right) datasets. Black dots identify locations used to determine $v_{tol}^{(95)}$ for each curve.

or motion-capture uncertainty), and are not classified as steps. As the tolerance is increased, the number of footsteps identified, f_{est}, first rises but then quickly plateaus at a static maximum value representing the total number of footsteps in the sequence. A model that produces foot-sliding artefacts will require higher tolerance before reaching its maximum. If the tolerance is increased further, the estimated number of footsteps eventually begins to decrease as separate footsteps start to be merged.

Plots of f_{est} as a function of v_{tol} on held-out data are provided in Fig. 5; the human and dog motion clips used as the basis for these plots and for the associated analysis are available in the supplement. (MA is not included since no data was held out from its training.) The plots show that MG is able to stay close to NAT in both scenarios. QN, which only is available for the human data, generates slightly too many steps, but is otherwise close to the natural footstep profile. The quadruped data appears to be more challenging than the human data, with the peaked behaviour of the estimated number of footsteps f_{est} for RNN and VAE indicating less distinctive synthetic locomotion that is likely to exhibit substantial foot sliding. MG, in contrast, again shows an f_{est}-profile very similar to that of natural motion.

For each model, we incremented v_{tol} in small steps (1.0 cm/s for human, 0.3 cm/s for quadruped) and extracted the first tolerance value $v_{tol}^{(95)}$ that reached 95% of the maximum number of footsteps identified for that model in our evaluation. These points are shown as black dots on the curves in Fig. 5. The tolerance threshold $v_{tol}^{(95)}$ essentially measures the 95th percentile of foot sliding in the motion. The lower this is, the crisper the motion is likely to be.

Table 3 shows the total estimated number of footsteps, the speed threshold, and the mean and standard deviation of the duration of the steps for different systems when resynthesising the held-out data from the two datasets. We note that MG almost always is the model that most closely adheres to the ground truth behaviour. Especially interesting is that MoGlow matches not only the mean but also the standard deviation of the natural step durations. Such

Table 3. Results from the objective evaluations: total number of footsteps f_{est}, speed tolerance $v_{tol}^{(95)}$ (cm/s) for capturing 95% of steps, mean and standard deviation of step durations (s), and bone-length RMSE (cm). The number closest to its natural counterpart in each column is shown in bold.

	Human					Quadruped				
ID	f_{est}	$v_{tol}^{(95)}$	μ	σ	RMSE	f_{est}	$v_{tol}^{(95)}$	μ	σ	RMSE
NAT	297	5.0	0.31	0.26	-	290	3.2	0.61	0.71	-
RNN	328	8.0	0.39	0.39	1.7	216	2.6	0.72	1.05	2.3
VAE	**278**	7.0	0.35	0.30	1.7	277	**2.9**	**0.61**	0.90	2.0
QN	318	**5.0**	0.23	0.19	**0.07**	-	-	-	-	-
MG	**278**	**5.0**	**0.32**	**0.23**	0.50	**295**	**2.9**	0.57	**0.75**	**0.51**

behaviour might be expected from an accurate probabilistic model, whereas deterministic models, not having any randomness and thus no entropy, are fundamentally limited not to match the statistics of the natural distribution in all respects.

Since the task-agnostic models in the objective evaluation were trained on joint positions, bone lengths need not be conserved in model output. This can lead to bone-stretching artefacts, and joints may even fly apart; cf. Ling et al. [2020]. Fortunately, bone-length deviation is easy to quantify objectively. Table 3 reports the RMSE of bone length in cm, simultaneously averaged across all joints and time-frames in the test data. We see that the error is small, meaning that bone lengths in MG output are stable and consistent.

5.4 Probabilistic aspects and further experiments

Having evaluated motion quality in-depth across tasks, we now present evidence to validate the wide applicability and the probabilistic aspects of the model. To increase the relevance for computer-graphics applications, we here change the pose representation to joint angles and apply the synthesised motion to a skinned character. We note that another option for obtaining skinned characters would be to train on joint positions in a skeleton with virtual joints like in Smith et al. [2019], and then apply inverse kinematics to recover joint angles, although this would add another computational step.

We created a new MoGlow model designed to investigate the ability of the method to learn from diverse motion data and reproduce its distribution. For this model, we constructed a new dataset by pooling the LaFAN1 dataset from Harvey et al. [2020], along with the Kinematica dataset.[6] We excluded trials involving wall and obstacle interaction as well as dancing, falling, stumbling, fighting, and sitting or lying on the ground. Nonetheless, this new data contains more varied motion than the data from Sec. 4.1, including crouching, hopping, walking while aiming, etc. This yielded a total of 1 h of data at 20 Hz (augmented to 4 h as before). All motion was retargeted to a uniform skeleton and the joint angles were converted to exponential maps [Grassia 1998]. The hips were expressed local to the floor-projected root, similar to before. For the new model, data dropout was reduced to 60%, which proved to generate smooth motion without losing adherence to the control. During synthesis, the raw model output was applied directly to the character, without any post-processing such as foot stabilisation.

[6]The data is available at https://github.com/ubisoft/Ubisoft-LaForge-Animation-Dataset and at https://github.com/Unity-Technologies/Kinematica_Demo, respectively.

As shown in our presentation video and in Fig. 1, we find that MoGlow not only is able to learn to produce high-quality motion from the new data, but that model output also successfully reflects the diversity of the material, and random samples of motion along the same path may take very different forms. MoGlow can thus produce a wide gamut of different motions for fixed control input, as expected for a strong probabilistic model under weak control signals. This is beneficial for increasing variation and naturalness, for example automatically generating sniffing behaviour when the dog is moving slowly. By training a similar model on all the human motion capture material, with no trials except climbing and running on walls excluded, even more varied output was produced, as shown at the very end of our presentation video.

In situations where greater control over motion diversity is desired, this may be obtained by reducing the sampling temperature or by using other, stronger control signals. For example, crouching or crawling motion might be consistently recovered without manual annotation of training data by training models where pelvic distance above ground is a control input instead of a model output.

Nothing about MoGlow is specific to locomotion. The generality of the approach is demonstrated by follow-up work [Alexanderson et al. 2020], performed after the locomotion studies described in this article but published before this article appeared, that shows that MoGlow successfully generalises to synthesising speech-driven gesture motion from speech acoustic features. Since gestures require time to prepare in order to be in synchrony with speech, it was necessary to provide that model with 1 second of future speech. That article also investigates style control of the output motion, which provides another option for constraining motion diversity.

5.5 Drawbacks and limitations

While being a powerful machine-learning method, MoGlow comes with some disadvantages of note in computer-graphics scenarios. Aside from the fact that machine learning affords less direct control over motion than hand animation does (and thus is more suited to high-level style control as mentioned in Sec. 5.4), the most relevant limitations relate to resource use at training and synthesis time.

Training a model like MoGlow demands substantial amounts of data and computation. In many graphics applications, waiting several hours to obtain an updated model is undesirable. Iteration time during model development may be sped up by training on multiple GPUs and by using model-surgery techniques [OpenAI et al. 2019] to avoid re-training new architectures from scratch. As for data, the various training and validation curves reported in Alexanderson and Henter [2020] suggest that the MG systems in this article are "data-limited", and that more training data should improve held-out data likelihood. Aside from recording additional material or pre-training on other motion databases, one might use high-quality data-augmentation techniques like those in Lee et al. [2018] to increase training-set size. This can be seen as a way to inject domain knowledge into the model-creation process.

MoGlow requires that frames are generated in sequence. Since the method describes an entire distribution of plausible poses, models furthermore tend to be deep and large. These properties may complicate interactive applications such as games. In general, it is easier to make good models fast than it is to make fast models good, and we expect it to be entirely possible to speed up MoGlow generation, e.g., using density distillation techniques like Huang et al. [2020] to create shallower models with similar accuracy as deeper ones. To compress the model footprint, neural-network pruning techniques like those surveyed in Blalock et al. [2020] are a compelling choice.

While MoGlow has performed well on the various motion tasks we have tried it on, we note that it does not contain any explicit physics model. We have seen rare instances of physically inappropriate motion, such as leaning stances where a real character would fall over. Reverse-time augmentation, when used, can give similar issues such as leaning forwards when running backwards at speed. We expect that these issues can be mitigated by more training data (reducing the need for augmentation), and by providing contact information as an input signal, but it might be more efficient to consider methods for introducing physics directly into the model. MoGlow also does not contain any model of human behaviour and intent, so in the absence of external information to guide the choice of behaviour, model output may switch between diverse locomotion modes and styles in an unstructured way.

6 CONCLUSION AND FUTURE WORK

We have described the first model of motion-data sequences based on normalising flows. This paradigm is attractive because flows 1) are probabilistic (unlike many established motion models), 2) utilise powerful, implicitly defined distributions (like GANs, but unlike classical autoregressive models), yet 3) are trained to directly maximise the exact data likelihood (unlike GANs and VAEs). Our model uses both autoregression and a hidden state (recurrence) to generate output sequentially, and incorporates a control scheme without algorithmic latency. (Non-causal control is a straightforward extension.) To our knowledge, no other Glow-based sequence models combine these desirable traits, and no other such model has incorporated hidden states, nor data dropout for more consistent control. Moreover, our approach is probabilistic from the ground up and generates convincing samples without entropy-reduction schemes like those in Brock et al. [2019]; Greenwood et al. [2017a,b]; Henter and Kleijn [2016]. Experimental evaluations show that the model produces high-quality synthetic locomotion for both bipedal and quadrupedal motion-capture data, despite their disparate morphologies. Subjective and objective results show that our proposal significantly outperforms task-agnostic LSTM and VAE-based approaches, coming close to natural motion recordings and performing on par with task-specific state-of-the-art locomotion models.

In light of the quality of the synthesised motion and the generally-applicable nature of the approach, we believe that models based on normalising flows can prove valuable for a wide variety of tasks incorporating motion data. Future work includes applying the method to additional tasks and domains, and making models lighter and faster for applied scenarios. Since models based on normalising flows allow exact and tractable inference, another interesting application would be to use the probabilities inferred by these models to also enable classification.

ACKNOWLEDGMENTS

This research was partially supported by Swedish Research Council proj. 2018-05409 (StyleBot), Swedish Foundation for Strategic Research contract no. RIT15-0107 (EACare), and by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

REFERENCES

Simon Alexanderson and Gustav Eje Henter. 2020. Robust model training and generalisation with Studentising flows. In *Proceedings of the Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models (INNF+'20, Vol. 2)*. Article 15, 9 pages. https://arxiv.org/abs/2006.06599

Simon Alexanderson, Gustav Eje Henter, Taras Kucherenko, and Jonas Beskow. 2020. Style-controllable speech-driven gesture synthesis using normalising flows. *Comput. Graph. Forum* 39, 2 (2020), 487–496. https://doi.org/10.1111/cgf.13946

Okan Arikan and David A. Forsyth. 2002. Interactive motion generation from examples. *ACM Trans. Graph.* 21, 3 (2002), 483–490. https://doi.org/10.1145/566570.566606

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NIPS'15)*. Curran Associates, Inc., Red Hook, NY, USA, 1171–1179. http://papers.nips.cc/paper/5956-scheduled-sampling-for-sequence-prediction-with-recurrent-neural-networks

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. 2020. What is the state of neural network pruning?. In *Proceedings of the Conference on Machine Learning and Systems (MLSys'20)*. 129–146. https://proceedings.mlsys.org/book/2020/hash/d2ddea18f00665ce8623e36bd4e3c7c5

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning (CoNLL'16)*. ACL, Berlin, Germany, 10–21. https://doi.org/10.18653/v1/K16-1002

Matthew Brand and Aaron Hertzmann. 2000. Style machines. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'00)*. ACM Press/Addison-Wesley Publishing Co., USA, 183–192. https://doi.org/10.1145/344779.344865

Christoph Bregler. 1997. Learning and recognizing human dynamics in video sequences. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*. IEEE Computer Society, Los Alamitos, CA, USA, 568–574. https://doi.org/10.1109/CVPR.1997.609382

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large scale GAN training for high fidelity natural image synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR'19)*. 35. https://openreview.net/forum?id=B1xsqj09Fm

Judith Bütepage, Michael J. Black, Danica Kragic, and Hedvig Kjellström. 2017. Deep representation learning for human motion prediction and classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'17)*. IEEE Computer Society, Los Alamitos, CA, USA, 1591–1599. https://doi.org/10.1109/CVPR.2017.173

Jinxiang Chai and Jessica K. Hodgins. 2005. Performance animation from low-dimensional control signals. *ACM Trans. Graph.* 24, 3 (2005), 686–696. https://doi.org/10.1145/1073204.1073248

Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2017. Variational lossy autoencoder. In *Proceedings of the International Conference on Learning Representations (ICLR'17)*. 17. https://openreview.net/forum?id=BysvGP5ee

CMU Graphics Lab. 2003. Carnegie Mellon University motion capture database. http://mocap.cs.cmu.edu/

Chris Cremer, Xuechen Li, and David Duvenaud. 2018. Inference suboptimality in variational autoencoders. In *Proceedings of the International Conference on Machine Learning (ICML'18)*. PMLR, 1078–1086. http://proceedings.mlr.press/v80/cremer18a.html

Gustavo Deco and Wilfried Brauer. 1994. Higher order statistical decorrelation without information loss. In *Advances in Neural Information Processing Systems (NIPS'94)*. MIT Press, Cambridge, MA, USA, 247–254. https://papers.nips.cc/paper/901-higher-order-statistical-decorrelation-without-information-loss

Chuang Ding, Pengcheng Zhu, and Lei Xie. 2015. BLSTM neural networks for speech driven head motion synthesis. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'15)*. ISCA, Grenoble, France, 3345–3349. https://www.isca-speech.org/archive/interspeech_2015/i15_3345.html

Laurent Dinh, David Krueger, and Yoshua Bengio. 2015. NICE: Non-linear independent components estimation. In *Proceedings of the International Conference on Learning Representations, Workshop Track (ICLR'15)*. 13. https://arxiv.org/abs/1410.8516

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2017. Density estimation using Real NVP. In *Proceedings of the International Conference on Learning Representations (ICLR'17)*. 32. https://openreview.net/forum?id=HkpbnH9lx

Ylva Ferstl, Michael Neff, and Rachel McDonnell. 2019. Multi-objective adversarial gesture generation. In *Proceedings of the ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG'19)*. ACM, New York, NY, USA, Article 3, 10 pages. https://doi.org/10.1145/3359566.3360053

Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. 2015. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'15)*. IEEE Computer Society, Los Alamitos, CA, USA, 4346–4354. https://doi.org/10.1109/ICCV.2015.494

Ian Goodfellow. 2016. NIPS 2016 tutorial: Generative adversarial networks. arXiv:1701.00160

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS'14)*. Curran Associates, Inc., Red Hook, NY, USA, 2672–2680. http://papers.nips.cc/paper/5423-generative-adversarial-nets

F. Sebastian Grassia. 1998. Practical parameterization of rotations using the exponential map. *J. Graph. Tools* 3, 3 (1998), 29–48. https://doi.org/10.1080/10867651.1998.10487493

Alex Graves. 2013. Generating sequences with recurrent neural networks. arXiv:1308.0850

David Greenwood, Stephen Laycock, and Iain Matthews. 2017a. Predicting head pose from speech with a conditional variational autoencoder. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'17)*. ISCA, Grenoble, France, 3991–3995. https://doi.org/10.21437/Interspeech.2017-894

David Greenwood, Stephen Laycock, and Iain Matthews. 2017b. Predicting head pose in dyadic conversation. In *Proceedings of the International Conference on Intelligent Virtual Agents (IVA'17)*. Springer, Cham, Switzerland, 160–169. https://doi.org/10.1007/978-3-319-67401-8_18

Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popović. 2004. Style-based inverse kinematics. *ACM Trans. Graph.* 23, 3 (2004), 522–531. https://doi.org/10.1145/1015706.1015755

Ikhansul Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and Taku Komura. 2017. A recurrent variational autoencoder for human motion synthesis. In *Proceedings of the British Machine Vision Conference (BMVC'17)*. BMVA Press, Durham, UK, Article 119, 12 pages. https://doi.org/10.5244/C.31.119

Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. 2020. Robust motion in-betweening. *ACM Trans. Graph.* 39, 4, Article 60 (2020), 12 pages. https://doi.org/10.1145/3386569.3392480

Dai Hasegawa, Naoshi Kaneko, Shinichi Shirakawa, Hiroshi Sakuta, and Kazuhiko Sumi. 2018. Evaluation of speech-to-gesture generation using bi-directional LSTM network. In *Proceedings of the ACM International Conference on Intelligent Virtual Agents (IVA'18)*. ACM, New York, NY, USA, 79–86. https://doi.org/10.1145/3267851.3267878

Gustav Eje Henter and W. Bastiaan Kleijn. 2016. Minimum entropy rate simplification of stochastic processes. *IEEE T. Pattern Anal.* 38, 12 (2016), 2487–2500. https://doi.org/10.1109/TPAMI.2016.2533382

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the International Conference on Learning Representations (ICLR'16)*. 22. https://openreview.net/forum?id=Sy2fzU9gl

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9, 8 (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Daniel Holden, Taku Komura, and Jun Saito. 2017. Phase-functioned neural networks for character control. *ACM Trans. Graph.* 36, 4, Article 42 (2017), 13 pages. https://doi.org/10.1145/3072959.3073663

Daniel Holden, Jun Saito, and Taku Komura. 2016. A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph.* 35, 4, Article 138 (2016), 11 pages. https://doi.org/10.1145/2897824.2925975

Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. 2015. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs (SA'15)*. ACM, New York, NY, USA, Article 18, 4 pages. https://doi.org/10.1145/2820903.2820918

Chin-Wei Huang, Faruk Ahmed, Kundan Kumar, Alexandre Lacoste, and Aaron Courville. 2020. Probability distillation: A caveat and alternatives. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI'20, Vol. 115)*. PMLR, 1212–1221. http://proceedings.mlr.press/v115/huang20c.html

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. 2018. Neural autoregressive flows. In *Proceedings of the International Conference on Machine Learning (ICML'18)*. PMLR, 2078–2087. http://proceedings.mlr.press/v80/huang18d.html

Ferenc Huszár. 2017. Is maximum likelihood useful for representation learning? http://www.inference.vc/maximum-likelihood-for-representation-learning-2/

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML'15)*. PMLR, 448–456. http:

//proceedings.mlr.press/v37/ioffe15.html
Lauri Juvela, Bajibabu Bollepalli, Junichi Yamagishi, and Paavo Alku. 2019. GELP: GAN-Excited Linear Prediction for Speech Synthesis from Mel-Spectrogram. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'19)*. ISCA, Grenoble, France, 694–698. https://doi.org/10.21437/Interspeech.2019-2008
Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aäron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. 2018. Efficient Neural Audio Synthesis. In *Proceedings of the International Conference on Machine Learning (ICML'18)*. PMLR, 2410–2419. http://proceedings.mlr.press/v80/kalchbrenner18a.html
Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. 2017. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Trans. Graph.* 36, 4, Article 94 (2017), 12 pages. https://doi.org/10.1145/3072959.3073658
Sungwon Kim, Sang-Gil Lee, Jongyoon Song, Jaehyeon Kim, and Sungroh Yoon. 2019. FloWaveNet: A generative flow for raw audio. In *Proceedings of the International Conference on Machine Learning (ICML'19)*. PMLR, 3370–3378. http://proceedings.mlr.press/v97/kim19b.html
Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR'15)*. 15. http://arxiv.org/abs/1412.6980
Diederik P. Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems (NeurIPS'18)*. Curran Associates, Inc., Red Hook, NY, USA, 10236–10245. http://papers.nips.cc/paper/8224-glow-generative-flow-with-invertible-1x1-con
Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations (ICLR'14)*. 14. http://arxiv.org/abs/1312.6114
Lucas Kovar and Michael Gleicher. 2004. Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph.* 23, 3 (2004), 559–568. https://doi.org/10.1145/1015706.1015760
Lucas Kovar, Michael Gleicher, and Frédéric Pighin. 2002. Motion graphs. *ACM Trans. Graph.* 21, 3 (2002), 473–482. https://doi.org/10.1145/566654.566605
Taras Kucherenko, Dai Hasegawa, Gustav Eje Henter, Naoshi Kaneko, and Hedvig Kjellström. 2019. Analyzing input and output representations for speech-driven gesture generation. In *Proceedings of the ACM International Conference on Intelligent Virtual Agents (IVA'19)*. ACM, New York, NY, USA, 97–104. https://doi.org/10.1145/3308532.3329472
Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. 2020. VideoFlow: A conditional flow-based model for stochastic video generation. In *Proceedings of the International Conference on Learning Representations (ICLR'20)*. 18. https://openreview.net/forum?id=rJgUfTEYvH
Neil Lawrence. 2005. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *J. Mach. Learn. Res.* 6, Nov. (2005), 1783–1816. http://www.jmlr.org/papers/v6/lawrence05a.html
Kyungho Lee, Seyoung Lee, and Jehee Lee. 2018. Interactive character animation by learning multi-objective control. *ACM Trans. Graph.* 37, 6, Article 180 (2018), 10 pages. https://doi.org/10.1145/3272127.3275071
Sergey Levine, Jack M. Wang, Alexis Haraux, Zoran Popović, and Vladlen Koltun. 2012. Continuous character control with low-dimensional embeddings. *ACM Trans. Graph.* 31, 4, Article 28 (2012), 10 pages. https://doi.org/10.1145/2185520.2185524
Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel van de Panne. 2020. Character controllers using motion VAEs. *ACM Trans. Graph.* 39, 4, Article 40 (2020), 12 pages. https://doi.org/10.1145/3386569.3392422
Peng Liu, Xixin Wu, Shiyin Kang, Guangzhi Li, Dan Su, and Dong Yu. 2019. Maximizing mutual information for Tacotron. arXiv:1909.01145
Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. 2018. Are GANs created equal? A large-scale study. In *Advances in Neural Information Processing Systems (NeurIPS'18)*. Curran Associates, Inc., Red Hook, NY, USA, 700–709. http://papers.nips.cc/paper/7350-are-gans-created-equal-a-large-scale-study
Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. 2018. Which training methods for GANs do actually converge?. In *Proceedings of the International Conference on Machine Learning (ICML'18)*. PMLR, 3481–3490. http://proceedings.mlr.press/v80/mescheder18a.html
Shakir Mohamed and Balaji Lakshminarayanan. 2016. Learning in implicit generative models. arXiv:1610.03483
Tomohiko Mukai and Shigeru Kuriyama. 2005. Geostatistical motion interpolation. *ACM Trans. Graph.* 24, 3 (2005), 1062–1070. https://doi.org/10.1145/1073204.1073313
Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber. 2007. *Documentation Mocap Database HDM05.* Technical Report CG-2007-2. Universität Bonn, Bonn, Germany. http://resources.mpi-inf.mpg.de/HDM05/07_MuRoClEbKrWe_HDM05.pdf
Kevin P. Murphy. 1998. *Switching Kalman Filters.* Technical Report 98-10. Compaq Cambridge Research Lab, Cambridge, MA, USA. https://www.cs.ubc.ca/~murphyk/Papers/skf.ps.gz
Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. 2019. Do deep generative models know what they don't know?. In *Proceedings of the International Conference on Learning Representations (ICLR'19)*. 19. https://openreview.net/forum?id=H1xwNhCcYm
OpenAI et al. 2019. Dota 2 with large scale deep reinforcement learning. arXiv:1912.06680
George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. 2019. Normalizing flows for probabilistic modeling and inference. arXiv:1912.02762
Dario Pavllo, David Grangier, and Michael Auli. 2018. QuaterNet: A quaternion-based recurrent model for human motion. In *Proceedings of the British Machine Vision Conference (BMVC'18)*. BMVA Press, Durham, UK, 14. http://www.bmva.org/bmvc/2018/contents/papers/0675.pdf
Vladimir Pavlović, James M. Rehg, and John MacCormick. 2000. Learning switching linear models of human motion. In *Advances in Neural Information Processing Systems (NIPS'00)*. MIT Press, Cambridge, MA, USA, 981–987. https://papers.nips.cc/paper/1892-learning-switching-linear-models-of-human-motion
Hai X. Pham, Yuting Wang, and Vladimir Pavlovic. 2018. Generative adversarial talking head: Bringing portraits to life with a weakly supervised neural network. arXiv:1803.07716
Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2019. WaveGlow: A flow-based generative network for speech synthesis. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'19)*. IEEE Signal Processing Society, Piscataway, NJ, USA, 3617–3621. https://doi.org/10.1109/ICASSP.2019.8683143
Albert Pumarola, Antonio Agudo, Aleix M. Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. 2018. GANimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*. Springer, Cham, Switzerland, 835–851. https://doi.org/10.1007/978-3-030-01249-6_50
Lawrence R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (1989), 257–286. https://doi.org/10.1109/5.18626
Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic back-propagation and approximate inference in deep generative models. In *Proceedings of the International Conference on Machine Learning (ICML'14)*. PMLR, 1278–1286. http://proceedings.mlr.press/v32/rezende14.html
Charles Rose, Michael F. Cohen, and Bobby Bodenheimer. 1998. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Comput. Graph.* 18, 5 (1998), 32–40. https://doi.org/10.1109/38.708559
Paul Rubenstein. 2019. Variational autoencoders are not autoencoders. http://paulrubenstein.co.uk/variational-autoencoders-are-not-autoencoders/.
Najmeh Sadoughi and Carlos Busso. 2018. Novel realizations of speech-driven head movements with generative adversarial networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'18)*. IEEE Signal Processing Society, Piscataway, NJ, USA, 6169–6173. https://doi.org/10.1109/ICASSP.2018.8461967
Najmeh Sadoughi and Carlos Busso. 2019. Speech-driven animation with meaningful behaviors. *Speech Commun.* 110 (2019), 90–100. https://doi.org/10.1016/j.specom.2019.04.005
Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. 2017. PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications. In *Proceedings of the International Conference on Learning Representations (ICLR'17)*. 10. https://openreview.net/forum?id=BJrFC6ceg
Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. 2018. Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'18)*. IEEE Signal Processing Society, Piscataway, NJ, USA, 4799–4783. https://doi.org/10.1109/ICASSP.2018.8461368
Harrison Jesse Smith, Chen Cao, Michael Neff, and Yingying Wang. 2019. Efficient neural networks for real-time motion style transfer. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2, 2, Article 13 (2019), 17 pages. https://doi.org/10.1145/3340254
Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. 2020. Local motion phases for learning multi-contact character movements. *ACM Trans. Graph.* 39, 4, Article 54 (2020), 14 pages. https://doi.org/10.1145/3386569.3392450
Supasorn Suwajanakorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. 2017. Synthesizing Obama: Learning lip sync from audio. *ACM Trans. Graph.* 36, 4, Article 95 (2017), 13 pages. https://doi.org/10.1145/3072959.3073640
Hideyuki Tachibana, Katsuya Uenoyama, and Shunsuke Aihara. 2018. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'18)*. IEEE Signal Processing Society, Piscataway, NJ, USA, 4784–4788. https://doi.org/10.1109/ICASSP.2018.8461829

Graham W. Taylor and Geoffrey E. Hinton. 2009. Factored conditional restricted Boltzmann machines for modeling motion style. In *Proceedings of the International Conference on Machine Learning (ICML'09)*. 1025–1032. https://icml.cc/Conferences/2009/papers/178.pdf

Graham W. Taylor, Geoffrey E. Hinton, and Sam T. Roweis. 2011. Two distributed-state models for generating high-dimensional time series. *J. Mach. Learn. Res.* 12, 28 (2011), 1025–1068. http://jmlr.org/papers/v12/taylor11a.html

Sarah Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews. 2017. A deep learning approach for generalized speech animation. *ACM Trans. Graph.* 36, 4, Article 93 (2017), 11 pages. https://doi.org/10.1145/3072959.3073699

Benigno Uria, Iain Murray, Steve Renals, Cassia Valentini-Botinhao, and John Bridle. 2015. Modelling acoustic feature dependencies with artificial neural networks: Trajectory-RNADE. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'15)*. IEEE Signal Processing Society, Piscataway, NJ, USA, 4465–4469. https://doi.org/10.1109/ICASSP.2015.7178815

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio. arXiv:1609.03499

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NIPS'17)*. Curran Associates, Inc., Red Hook, NY, USA, 6306–6315. http://papers.nips.cc/paper/7210-neural-discrete-representation-learning

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS'17)*. Curran Associates, Inc., Red Hook, NY, USA, 5998–6008. https://papers.nips.cc/paper/7181-attention-is-all-you-need

Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. 2018. End-to-end speech-driven facial animation with temporal GANs. In *Proceedings of the British Machine Vision Conference (BMVC'18)*. BMVA Press, Durham, UK, 12. http://www.bmva.org/bmvc/2018/contents/papers/0539.pdf

Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. 2020. Realistic speech-driven facial animation with GANs. *Int. J. Comput. Vis.* 128, 5 (2020), 1398–1413. https://doi.org/10.1007/s11263-019-01251-8

Jack M. Wang, David J. Fleet, and Aaron Hertzmann. 2008. Gaussian process dynamical models for human motion. *IEEE T. Pattern Anal.* 30, 2 (2008), 283–298. https://doi.org/10.1109/TPAMI.2007.1167

Xin Wang, Shinji Takaki, and Junichi Yamagishi. 2018. Autoregressive neural f0 model for statistical parametric speech synthesis. *IEEE/ACM T. Audio Speech* 26, 8 (2018), 1406–1419. https://doi.org/10.1109/TASLP.2018.2828650

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. 2017. Tacotron: Towards end-to-end speech synthesis. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH'17)*. ISCA, Grenoble, France, 4006–4010. https://doi.org/10.21437/Interspeech.2017-1452

Zhiyong Wang, Jinxiang Chai, and Shihong Xia. 2019. Combining Recurrent Neural Networks and Adversarial Training for Human Motion Synthesis and Control. *IEEE T. Vis. Comput. Gr.* (2019), 14. https://doi.org/10.1109/TVCG.2019.2938520

Greg Welch and Gary Bishop. 1995. *An Introduction to the Kalman Filter*. Technical Report 95-041. Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA. https://techreports.cs.unc.edu/papers/95-041.pdf

Youngwoo Yoon, Woo-Ri Ko, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. 2019. Robots learn social skills: End-to-end learning of co-speech gesture generation for humanoid robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'19)*. IEEE Robotics and Automation Society, Piscataway, NJ, USA, 4303–4309. https://doi.org/10.1109/ICRA.2019.8793720

G. Udny Yule. 1927. On a method of investigating periodicities disturbed series, with special reference to Wolfer's sunspot numbers. *Philos. T. R. Soc. Lond.* 226, 636–646 (1927), 267–298. https://doi.org/10.1098/rsta.1927.0007

Heiga Zen and Andrew Senior. 2014. Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'14)*. IEEE Signal Processing Society, Piscataway, NJ, USA, 3844–3848. https://doi.org/10.1109/ICASSP.2014.6854321

Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. 2019. Fixup initialization: Residual learning without normalization. In *Proceedings of the International Conference on Learning Representations (ICLR'19)*. 16. https://openreview.net/forum?id=H1gsz30cKX

He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. 2018. Mode-adaptive neural networks for quadruped motion control. *ACM Trans. Graph.* 37, 4, Article 145 (2018), 11 pages. https://doi.org/10.1145/3197517.3201366

Yi Zhou, Zimo Li, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. 2018. Auto-conditioned recurrent networks for extended complex human motion synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR'18)*. 13. https://openreview.net/forum?id=r11Q2SlRW