Transferring Movement Understanding for Parkinson's Therapy by Generative Pre-Training

Emily Napier[λ∓ *†] **Gavia Gray**[λ∓ ‡ †] **Tristan Loria**[§] **Veronica Vuong**[§]

Michael H. Thaut[§] **Sageev Oore**[λ∓]

Dalhousie University[λ]
Vector Institute[∓]
University of Toronto[§]

Abstract

Motion data is a modality of clinical importance for Parkinson's research but modeling it typically requires careful design of the machine learning system. Inspired by recent advances in autoregressive language modeling, we investigate the extent to which these modeling assumptions may be relaxed. We quantize motion capture data into discrete tokens and apply a generic autoregressive model to learn a model of human motion. Representing both positions and joint angles in a combined vocabulary, we model forward and inverse kinematics in addition to autoregressive prediction in 3D and angular space. This lets us pre-train on a 1B token, 40 hour dataset of motion capture, and then finetune on one hour of clinically relevant data in a downstream task. Despite the naivety of this approach, the model is able to perform clinical tasks and we demonstrate high performance classifying 5 hours of dance data.

1 Introduction

Recent advances in large language models have demonstrated methods for pre-training a model on a large dataset and using this same model in many downstream tasks [Devlin et al., 2019]. This includes classifying data from a few examples where the model must already understand the domain over which one intends to train the classifier. Therefore, if human motion can be interpreted similarly to how these models interpret language, training a language model on a dataset of generic human motion could improve performance on downstream tasks, freeing the practitioner from designing custom machine learning approaches in each setting.

Motion data in healthcare will always inevitably be in limited supply, if not because of the cost of collecting it, then because of patient privacy concerns. We demonstrate that pre-training language models on unlabeled motion-capture data may follow similar scaling principles as text datasets. This could reduce the need for clinical data, as a model may learn dynamics of human motion prior to solving specific problems in a clinical task setting.

Our main contribution here is an exploration of the performance of pre-trained language models on motion data with a minimum of modality-specific assumptions, paving the way for further work in this area translating progress from language modeling. Data is encoded generically as text using uniform

[*] emily.napier@dal.ca
[†] Equal contribution
[‡] gngdb.labs@gmail.com

1st Workshop on Deep Generative Models for Health at NeurIPS 2023.

quantization, while multi-task training—inspired by T5 [Raffel et al., 2019]—supports efficient learning from that text. Following T5, we name the resulting model *Multipurpose Motion to Motion Multitask Model (M5)* as we demonstrate it to be useful in transfer learning tasks. Performance on clinical and non-clinical downstream tasks is investigated in Section 5.

2 Related Work

Deep learning has been explored to model motion, but many prior methods specialize the model to the modality leading to continuous models that make explicit choices for how to model, for example, time and space [Valle-Pérez et al., 2021, Aksan et al., 2019, 2020, Zhu et al., 2023]. The architecture applied in this work aims to avoid significant specialization, by treating the data as a long sequence of discrete tokens, inspired by work in offline reinforcement learning (e.g.,Janner et al. [2021]). This uniform quantization method is generic and not learned, in contrast to vector quantisation methods [Lucas* et al., 2022, Zhang et al., 2023]. Other methods pair motion with additional modalities Zhang et al. [2023], Valle-Pérez et al. [2021] such as text or music to broaden the scope of motion understanding, while our model trains on multiple kinematics tasks and representations of pose.

PoseGPT [Lucas* et al., 2022] and *MotionGPT* [Zhang et al., 2023] are closest to this work, as both involve training a generic autoregressive transformer on tokens representing motion. However:

- Both employ VQ-VAE [van den Oord et al., 2018] autoencoders to map from the continuous space of poses to a discrete latent sequence of tokens. In this work, we apply a uniform quantization that does not need to be learned. This work can be viewed as an ablation of the vector quantization used by *PoseGPT* and *MotionGPT*.
- *PoseGPT* conditions on an action label and *MotionGPT* conditions on text, whereas this work focuses only on uncontional autoregressive modeling.
- This model trains on a variety of sequence tasks, described below, but including forward and inverse kinematics and causal modeling on both positional and angular representations of pose.

3 Methods

We focus on GPT-based [Radford et al., 2019] causal transformers building on minGPT [Karpathy]. Transformers have demonstrated improved performance [Vaswani et al., 2017] over previous sequence modeling architectures, especially in natural language processing. By casting the learning problem in discrete tokens, it is possible to add reserved tokens for any downstream task required. The mask allows each of these tasks to attend to whichever indices are required, e.g., classification tasks typically attend to the entire context.

The pretraining phase is the process during which the model learns the motion-capture domain, comprised of tasks including forward kinematics, inverse kinematics, and autoregressive kinematics prediction. This phase is carried out using the AMASS dataset [Mahmood et al., 2019], comprising 40 hours of motion capture data in a coherent 24 joint format [Loper et al., 2015]. We hypothesize, based on T5's results [Raffel et al., 2019], that a model trained on a larger corpus of kinematics data combinations will have a better understanding of the domain, and will better translate that information to downstream tasks [Devlin et al., 2019].

To convert motion-capture data to a format that can be used in any language model architecture, we use uniform quantization. This strategy creates a vocabulary of tokens that describe joint angle or joint position as distinct sets of characters. The model can convert between the two dictionaries, and models the relationship between tokens in each space. The process of tokenization is described in detail in Appendix A.

3.1 Causal Model Training

Causal modeling typically refers to the practice of training a model to autoregressively predict the next token [Gregor et al., 2013]. At test time, the final predicted token can be appended to the original sequence to generate novel outputs. Our model learns autoregressive joint angle and position

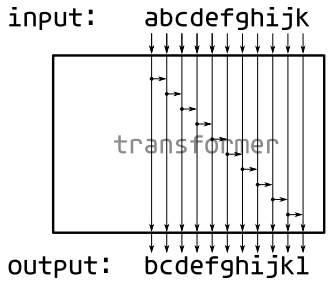


Figure 1: A diagram describing the causal relationships between tokens in the autoregressive transformer models used in this work. Tokens at the output (bottom) only depend on previous tokens on the offset sequence at the input (top).

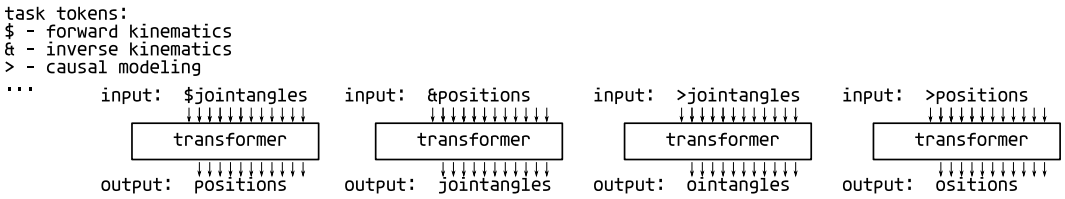


Figure 2: Illustration of how the transformer operates on an example of text with conditioning tokens indicating the task prepended to the sequence. Each task is approached autoregressively, and the model is trained to solve all tasks in parallel. Transformers here are causal as described in Figure 1.

prediction tasks with a causal mask as described above, and learns forward and inverse kinematics tasks with the same causal mask by training autoregressively on a sequence that interleaves frames from each modality as shown in Figure 2.

Empirical results have found language models follow scaling laws [Kaplan et al., 2020] that relate the number of tokens in the dataset to the number of parameters required by the model. In Section 4.1, we explore the scaling relationship with the AMASS dataset, containing $<$ 1B tokens. Models trained were GPT-like [Radford et al., 2019] decoder-only transformers building on minGPT [Karpathy]. Parameter counts ranged from 150,000 to 26 million. 26 million is approximately equal to the proportion suggested by Hoffmann et al. [2022]($\sim$20 times fewer parameters than the tokens in the dataset at this scale).

4 Pretraining Results

In this section we detail the performance of the pretrained model on the AMASS dataset.

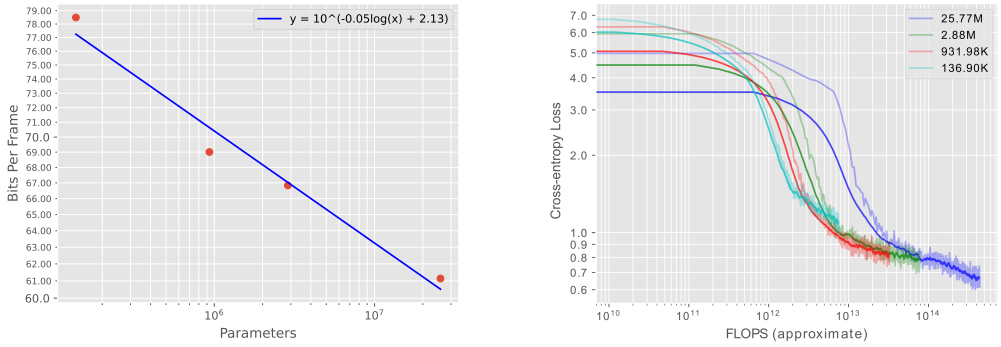
The model was trained on the following tasks:

- Predicting the next token:
  - For 3D joint angles, here called *Angular Causal Modeling*
  - For positions in 3D space, here called *Positional Causal Modeling*
- Inferring the 3D position of joints from angles, here called *Forward Kinematics*
- Inferring the 3D joint angles from positions, here called *Inverse Kinematics*

Our target downstream task on the clinical dataset was classification of positions in 3D space. Table 1 shows the pre-training results of each task combination that includes the *Positional Causal Modeling* task. The model pre-trained on all tasks achieved 92.7% +/- 1.9% accuracy predicting the next token versus 58.1% by the model trained on the *Positional Causal Modeling* task only. A complete characterisation of the motion modeling results can be found in Appendix C.

Table 1: Comparison of how the model performs on the *Positional Causal Modeling* task based on the pre-training tasks. The final cross-entropy loss is computed after training on 1e9 *Positional Causal Modeling* (PCM) tokens.

Pre-training Tasks	PCM Cross-Entropy Loss
PCM	0.6676
PCM, IK	0.6552
PCM, IK, FK	0.6535
All tasks	0.6314



(a) Model size scaling law.

(b) FLOPs learning curve.

Figure 3: Figure 3a illustrates a linear scaling law in motion data, between the number of parameters and the compression rate (bits per frame). Figure 3b demonstrates learning curves at different model scales showing convergence of a trend against flops spent.

4.1 A Scaling Law for Quantized Motion Modeling

Prior work training language models on text has established that performance of the model may be inferred from the number of tokens and the number of parameters allocated for training that model [Hestness et al., 2017, Kaplan et al., 2020, Hoffmann et al., 2022]. The relationship depends on the modality; we should not expect the same scaling law to apply on motion data. Experimentally, we find that the cross-entropy loss (also measured here in bits per frame) is inversely linearly proportional to the model size on a log scale, as expected for a scaling law with the precise relationship observed illustrated in Figure 3 and Appendix B.

5 Fine-tuning Results

We fine-tune the model on a dataset of 2-dimensional joint positions taken from videos of Parkinson's patient movement Li et al. [2018a]. In order to fine-tune the model on this dataset 2D joint positions were converted to match our 3D joint position vocabulary, and scale the range of the y and z dimensions of the Parkinson's dataset to the 10-90th percentile of the AMASS dataset.

Performance on the associated clinical tasks is summarised in Table 2 with complete results described in Appendix D. We observed poor performance of *M5* versus the kinematic random forest of Li et al. [2018a]. The likely reason for this is the small movements present in some tasks, for example the communication task involved patients sitting and talking. Quantization made discriminating movements below approximately 1cm impossible. The drinking task involved a larger movement and was likely easier to distinguish for this reason.

It's also possible that the preprocessing to tokenize the 2D motion capture data may have made performance difficult for a model pretrained on 3D data. The relative performance of the *M5* from scratch (*M5-fs*) model suggests that the data has left the domain of motion the model saw during training, because the randomly initialized model is sometimes able to reach the performance of *M5* pre-trained (*M5-pt*).

Table 2: Performance on AIST++ and Parkinsons [Li et al., 2018a] tasks. *M5-pt* refers to the pre-trained model and *M5-fs* refers to a model trained from scratch. In this table we recreate the methods used by [Li et al., 2018a] on the AIST++ dataset as a point of comparison.

Task	Comm.	Drinking	Leg Agility	Multiclass	UDsysRS	UPDRS
Metric	AUC	AUC	AUC	Accuracy		
Li et al. [2018b]	**0.93**	0.63	**0.77**	**71.4 %**	**2.91**	**7.76**
M5-pt	0.76 +/- 0.16	**0.72 +/- 0.14**	0.54 +/- 0.10	68.3 +/- 14.1%	3.23 +/- 1.2	10.59 +/- 3.8
M5-fs	0.79 +/- 0.12	0.66 +/- 0.14	0.66 +/- 0.18	64.3 +/- 12.5%	3.44 +/- 1.3	10.61 +/- 4.0

Task	Genre	Situation	Dancer
Metric	Accuracy	Accuracy	Accuracy
Li et al. [2018b]	82.9 %	92.9 %	68.6 %
M5-pt	**89.1 %**	**97.1 %**	**80.7 %**

To investigate the performance on a dataset with a larger movement range and a larger number of subjects, we also compared the performance of *M5* and the random forest of Li et al. [2018a] on the AIST++ [Li et al., 2021] dataset. The results support the hypothesis that the granularity of quantization is an issue, as *M5* performs better on a task involving larger movements. In addition, the AIST++ dataset includes more hours of total motion capture allowing a more reliable estimate of the performance.

6 Conclusion

Relaxing the assumptions present in modeling motion permits training generic autoregressive models and adapting innovations in text modeling. *M5* demonstrates good performance despite a limited representation of the data and allows the exploration of the scaling properties of motion modeling. However, coarse quantization likely fails the model in the clinical task examined. This suggests that future work would benefit from improvements to tokenize fine motor signals in the motion data. As the volume of publicly available motion capture data grows, the scaling results presented here suggest that the performance of generic autoregressive models for transfer learning tasks in clinical settings will become increasingly relevant.

Acknowledgments and Disclosure of Funding

Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, companies sponsoring the Vector Institute, and NSERC.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019. URL `https://arxiv.org/abs/1910.10683`.

Guillermo Valle-Pérez, Gustav Eje Henter, Jonas Beskow, André Holzapfel, Pierre-Yves Oudeyer, and Simon Alexanderson. Transflower: probabilistic autoregressive dance generation with multimodal attention. 2021.

Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. Structured prediction helps 3d human motion modelling. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. First two authors contributed equally.

Emre Aksan, Peng Cao, Manuel Kaufmann, and Otmar Hilliges. Attention, please: A spatio-temporal transformer for 3d human motion prediction. *CoRR*, abs/2004.08692, 2020. URL `https://arxiv.org/abs/2004.08692`.

Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. Motionbert: A unified perspective on learning human motion representations, 2023.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem, 2021. URL https://arxiv.org/abs/2106.02039.

Thomas Lucas*, Fabien Baradel*, Philippe Weinzaepfel, and Grégory Rogez. PoseGPT: Quantization-based 3d human motion generation and forecasting. In *European Conference on Computer Vision (ECCV)*, 2022.

Yaqi Zhang, Di Huang, Bin Liu, Shixiang Tang, Yan Lu, Lu Chen, Lei Bai, Qi Chu, Nenghai Yu, and Wanli Ouyang. Motiongpt: Finetuned llms are general-purpose motion generators, 2023.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

Andrej Karpathy. minGPT. https://github.com/karpathy/minGPT. Accessed: 2023-03-09.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, October 2019.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6): 248:1–248:16, October 2015.

Karol Gregor, Ivo Danihelka, Andriy Mnih, Charles Blundell, and Daan Wierstra. Deep autoregressive networks. 2013. doi: 10.48550/ARXIV.1310.8499. URL https://arxiv.org/abs/1310.8499.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically, 2017. URL https://arxiv.org/abs/1712.00409.

Michael H Li, Tiago A Mestre, Susan H Fox, and Babak Taati. Vision-based assessment of parkinsonism and levodopa-induced dyskinesia with pose estimation. *Journal of neuroengineering and rehabilitation*, 15(1):1–13, 2018a.

Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Learn to dance with AIST++: Music conditioned 3d dance generation, 2021.

Michael H. Li, Tiago A. Mestre, Susan H. Fox, and Babak Taati. Vision-based assessment of parkinsonism and levodopa-induced dyskinesia with pose estimation. *Journal of NeuroEngineering and Rehabilitation*, 15(1), nov 2018b. doi: 10.1186/s12984-018-0446-z. URL https://doi.org/10.1186%2Fs12984-018-0446-z.

Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, Piscataway, NJ, USA, July 2017. IEEE.

Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. In *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, page 299, 2018. URL `http://bmvc2018.org/contents/papers/0675.pdf`.

Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines, 2016. URL `https://arxiv.org/abs/1602.00134`.

A Data Tokenization

Motion data is typically modeled as a sequence of frames, wherein each frame is represented using a set of joint angles – referred to as a pose. In the notation of SMPL [Loper et al., 2015] a complete frame of motion can be described by:

- $\vec{\theta}$: pose, composed of axis-angle 3D rotation vectors $\vec{\omega}$: $\vec{\theta} = [\vec{\omega}_0^T, ..., \vec{\omega}_K^T]$ for K angles.
- $\vec{\beta}$: shape parameters.

Using the shape parameters, the 3D positions in space occupied by each joint can be recovered by forward kinematics. The sequence of motion can therefore be represented as a sequence of joint angles or a sequence of 3D positions through time. In this paper, our model processes both representations without modification. This is achieved by quantizing both forms of data into a discrete set of tokens.

A.1 Tokenization

Motion data is typically a sequence of poses, each pose is a sequence of joint angles, typically the 24 canonical joints of the SMPL body model [Loper et al., 2015]. At time of writing, the largest publicly available dataset of human motion is the AMASS [Mahmood et al., 2019] dataset.

Following the method described in Janner et al. [2021] for tokenizing, the data is uniformly binned on each dimension of each joint axis-angle vector. In some experiments, where indicated, we trained the model on a reduced set of joints, these are shown in Figure 4. This was done to accelerate inference for demonstrations. The resulting integers are matched to arbitrary alphanumeric unicode characters so they can be used in a generic text model as is. Each frame is represented by a "word" with a space placed between frames.

Uniform Quantization Let x be a continuous variable that we want to quantize, and let $q_1, q_2, ..., q_n$ be the n quantization levels or bins. We assume that the bins are uniformly spaced, so that the distance between adjacent bins is the same and can be denoted as Δ.

Then, the quantization operation $Q(x)$ can be defined as:

$$Q(x) = q_{k(x)} \quad \text{if} \quad q_{k(x)-1} \leq x < q_k(x)$$

where k is the index of the bin that contains x, and is given by:

$$k(x) = \left\lfloor \frac{x - q_1}{\Delta} \right\rfloor + 1$$

Here, $\lfloor \cdot \rfloor$ denotes the floor function, which rounds down to the nearest integer.

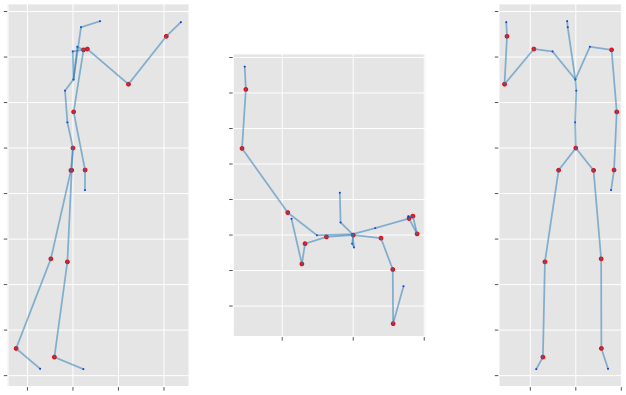


Figure 4: A subset of the 24 SMPL [Loper et al., 2015] joints are illustrated. These were used in some experiments to reduce frame character length.

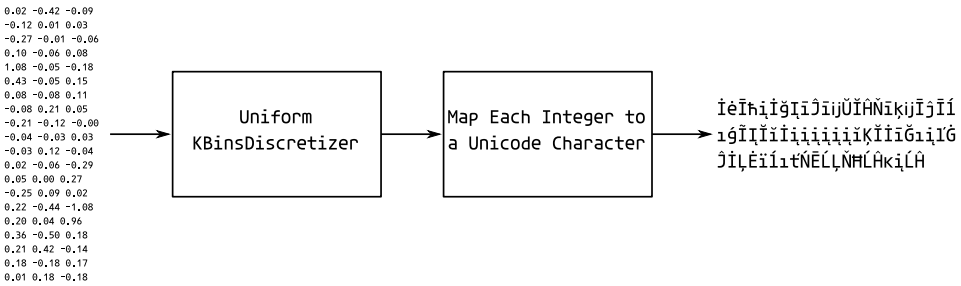


Figure 5: Flow chart describing the encoding process, the data is uniformly quantized and the resulting integers are mapped to unicode characters.

Note that the quantized value $Q(x)$ takes on one of the n possible quantization levels $q_1, q_2, ..., q_n$, depending on which bin x falls into.

After quantization we match the integer IDs of each quantized bin to a unique Unicode character so we may manipulate the data as text and place it in workflows designed for text. An example of this data is shown in Figure 6.

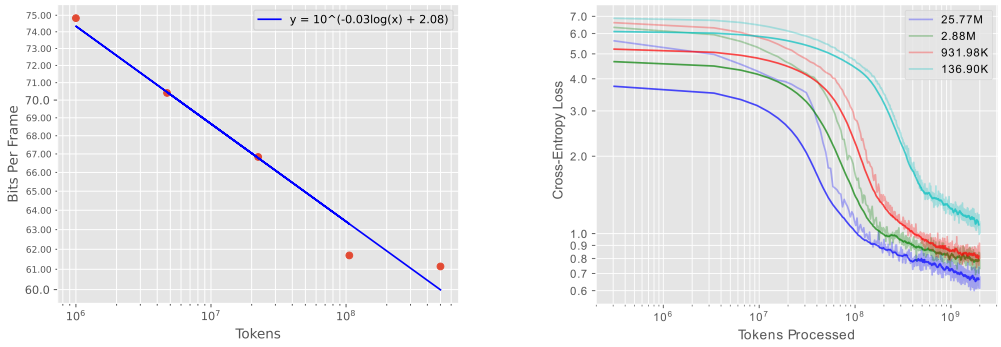
The full process for encoding is shown in Figure 5, input frames of motion pass in as sets of 3D axis-angle vectors. These are uniformly quantized and each integer is mapped to a unique unicode character. The position in this sequence of characters determines the frame and joint each character is associated with, as shown in Figure 6.

B Motion-capture data scaling laws

To further illustrate the scaling law shown in Figure 3, Figure 7 explores the relationship between compression rate and the number of tokens processed. As the number of tokens processed increases,

```
        [           frame              ] [      frame+1    ...
motion: ħıČĴĬşĭŀĽįlJĮįĮįįĮħēĉîĺðîòğĭĺłįĮįĮįĮ Þſ₫ĠựĪĢģlJĐıĴŢīīłĮĮĮ...
joint:  0  1  2  3  4  5  6  7  8  9 10 11  0  1  2  3  4  5   ...
frame:  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  1  1   ...
```

Figure 6: Example of motion data as encoded text. Joint and frame indexes were used in the positional embedding. The sequence illustrated here has 11 joints to reduce figure width.



(a) Number of Tokens

(b) Tokens Processed

Figure 7: The scaling law and learning curve are illustrated in terms of Tokens, in Figure 7a shows the scaling law relating the compression rate to the number of tokens processed and Figure 7b shows the same learning curves plotted in Figure 3 but the x-axis chosen in tokens processsed.

the compression rate achieved by the models is able to decrease further, with larger models able to make best use of additional data.

This relationship is illustrated in Figure 8, wherein each point in the scatter plot is a model trained with a different proportion of the dataset included. As the model size increases, spending more flops requires supplying the models with more data in order to continue to see improvements in the cross-entropy loss.

C Motion Modeling

All of these tasks are extracted from the AMASS dataset [Mahmood et al., 2019], which consists of joint angles. The 3D positions are produced using the SMPL body model [Loper et al., 2015]. Both are uniformly quantized into tokens for all tasks, as described in Section A.

Given two seconds of motion as context, this task is to predict the next 400ms of motion. This benchmark was introduced by Aksan et al. [2019] and we compare to their results and the results they replicated to compare against. An example of this prediction task for a single joint is illustrated in Figure 9. The quantization noise from discretizing the continuous joint angles is visible.

Quantization noise limits the performance this model is able to achieve on this task. To investigate this, we encoded and decoded the targets and substituted this as a prediction, to see what the best possible performance that an oracle could achieve, picking the correct token at every step. This is called *Quantized Oracle* in Table 3. This could be addressed by increasing the resolution of the quantization. However, this can become cumbersome, as the parameter cost is quadratic in the number of tokens the model encodes.

Transformer models for motion modeling exist in the literature but typically the data is trained on as continuous 3D joint angles. Rotations in 3D can be represented using at least six formalisms and errors between angles may be similarly computed in various ways. For example, geodesic error is the size of the minimum rotation in radians to rotate from one angular orientation to another. In Table 3 this metric is called *Joint Angle*.

The remaining metrics in Table 3 are described by Aksan et al. [2019]. Briefly:

- *Euler* is the RMSE between the joint angles expressed as Euler angles
- Two are defined over positions computed from the predicted joint angles using a predefined forward kinematics model defined by Aksan et al. [2019]:
  - *Positional* is the MSE between positions in 3D space
  - *PCK (AUC)* is ratio of joints within a spherical threshold around the target position in 3D space

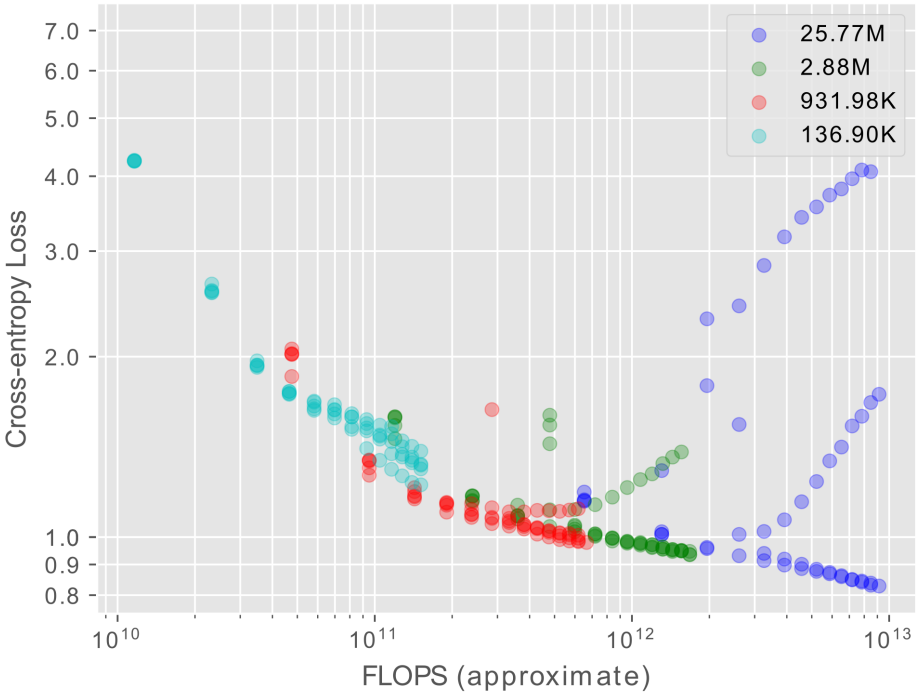
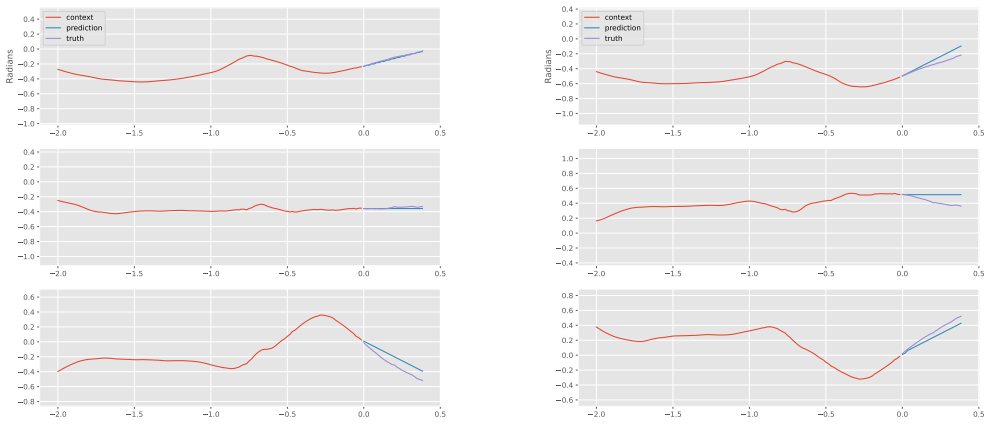


Figure 8: All dataset sizes are plotted overlaid as a scatter plot with model size annotated, each point is a model trained with a different proportion of the dataset included. The performance of the model scales with the number of FLOPs spent, as long as the model size increases and enough training data is available. The larger models can be seen to overfit and the test loss diverges when the dataset they are trained on is too small.



(a) Left Shoulder

(b) Right Shoulder

Figure 9: Example task from the autoregressive motion modeling task described in Appendix C, focusing only on the left and right shoulders.

Table 3: *AMASS results* comparing to the work of Aksan et al. [2020] and the results used for comparison in their paper in the style presented in their paper. ↓ indicates metrics where lower is better and ↑ indicates metrices where higher is better. * indicates the model was evaluated by Aksan et al. [2020] rather than the original authors, and the results of Aksan et al. [2020] are as they reported them. Our model is referred to here as *M5*.

	Euler ↓				Joint Angle ↓				Positional ↓				PCK (AUC) ↑			
milliseconds	100	200	300	400	100	200	300	400	100	200	300	400	100	200	300	400
Zero-Velocity [Martinez et al., 2017, Aksan et al., 2019]	1.91	5.93	11.36	17.78	0.37	1.22	2.44	3.94	0.14	0.48	0.96	1.54	0.86	0.83	0.84	0.82
Seq2seq [Martinez et al., 2017, Aksan et al., 2019]	2.01	5.99	11.22	17.33	0.37	1.17	2.27	3.59	0.14	0.45	0.88	1.39	0.86	0.84	0.85	0.83
QuaterNet [Pavllo et al., 2018, Aksan et al., 2019]	1.49	4.70	9.16	14.54	0.26	0.89	1.83	3.00	0.10	0.34	0.71	1.18	0.90	0.87	0.88	0.85
DCT-GCN (ST)* [Mao et al., 2019]	1.23	4.00	8.05	13.04	0.24	0.77	1.60	2.66	0.09	0.31	0.63	1.06	0.92	0.89	0.89	0.87
DCT-GCN (LT)* [Mao et al., 2019]	1.27	4.18	8.37	13.38	0.24	0.80	1.65	2.71	0.09	0.31	0.65	1.07	0.91	0.89	0.89	0.87
RNN-SPL [Aksan et al., 2019]	1.33	4.13	8.03	12.84	0.22	0.73	1.51	2.51	0.08	0.28	0.57	0.96	0.93	0.90	0.90	0.88
Transformer	1.30	4.01	7.88	12.69	0.22	0.73	1.52	2.54	0.08	0.28	0.58	0.97	0.92	0.90	0.90	0.88
ST-Transformer	1.11	3.61	7.31	12.04	0.20	0.68	1.45	2.48	0.08	0.27	0.57	0.97	0.93	0.90	0.90	0.88
Quantized Oracle	0.26	0.57	0.84	1.12	0.05	0.10	0.15	0.20	0.06	0.11	0.17	0.23	0.93	0.95	0.97	0.97
M5 (26M)	1.70	5.83	12.11	20.08	0.21	0.75	1.58	2.64	0.20	0.78	1.77	3.08	0.78	0.72	0.73	0.68

Table 4: *AMASS results* computing metrics to describe performance of the model in positional causal modeling, forward kinematics and inverse kinematics.

	Euler ↓				Joint Angle ↓				Positional ↓				PCK (AUC) ↑			
milliseconds	100	200	300	400	100	200	300	400	100	200	300	400	100	200	300	400
M5 (26M) FK									0.07	0.23	0.47	0.82	0.91	0.90	0.91	0.88
M5 (26M) IK	1.38	5.16	10.70	17.36	0.21	0.71	1.46	2.35	0.19	0.69	1.46	2.43	0.79	0.75	0.76	0.72
M5 (26M) PCM									0.08	0.30	0.65	1.15	0.90	0.87	0.88	0.84

Comparing against existing methods in Table 3 we can see that our model fails mainly predicting positions, scoring poorly on Positional and AUC. This may be expected, because this task only tests angular causal modeling. Some metrics for the remaining pretraining tasks are shown in Table 4. Angular performance is similar to the performance of these other published models, performing worse on the euler angle metric. The cross-entropy loss minimized during training operates on quantized axis-angle vectors and the geodesic error here is proportional to squared error between axis-angle vectors.

In Table 4 the forward and inverse kinematics are performing a different task. Both are set up with 1 second of interleaved frames of angles and positions encoded as text, then the task is infer the next missing frame of either angular or positional data. This is an easier task than inferring frames conditioned only on prior context, and forward kinematics in particular is a simple deterministic function. However, positional causal modeling is the same task as reported in Table 3 with the only difference being that the next token predicted describes a position in 3D space rather than an angle. The model is then able to match the results reported by Aksan et al. [2020] on the positional metrics which it was underperforming in Table 3.

We believe that demonstrating competitive autoregressive modeling despite the restrictions this language model is trained with, generically with no inductive biases about the modality, is a valuable result. In particular, this model is restricted to use the most generic form of discretization, each dimension of each angle is a separate token. A tokenization scheme that improves upon this in any way could outperform the methods listed in this paper.

D Parkinson's Severity Classification

Parkinson's patients can experience bradykinesia, characterized by slowness of movement. Patients treated with levodopa may develop levodopa induced dyskinesia (LID) resulting in dyskinesia characterized by involuntary movements. Li et al. [2018b] classified movements with a binary classification task identifying pathological movements, a regression task indicating the severity of symptoms, and a multi-class classification task identifying whether the patient movements have qualities of PD, LID, or normal movement. The dataset comes from Wei et al. [2016] wherein a convolutional neural network was trained to detect 2D joint positions.

We present our results after fine-tuning on our pre-trained model, and training on the same model architecture from scratch. Both models are trained simultaneously on all classification tasks, along

Table 5: Performance on binary classification of communication examples.

	Neck	Rarm	Larm	Trunk	Rleg	Lleg	Mean	Sigma
Li et al. [2018b] F1	0.941	0.920	0.929	0.960	0.819	0.865	**0.906**	
Li et al. [2018b] AUC	0.935	0.957	0.946	0.983	0.852	0.907	**0.930**	
Pretrained								
F1	0.590	0.533	0.671	0.605	0.535	0.707	0.607	0.148
AUC	0.712	0.729	0.785	0.763	0.751	0.831	0.762	0.156
From Scratch								
F1	0.549	0.562	0.675	0.575	0.537	0.691	0.598	0.198
AUC	0.827	0.771	0.810	0.805	0.766	0.773	0.792	0.118

Table 6: Performance on binary classification of drinking examples.

	Neck	Rarm	Larm	Trunk	Rleg	Lleg	Mean	Sigma
Li et al. [2018b] F1	0.711	0.148	0.289	0.643	0.594	0.617	0.500	
Li et al. [2018b] AUC	0.774	0.418	0.557	0.687	0.673	0.696	0.634	
Pretrained								
F1	0.474	0.522	0.601	0.536	0.544	0.637	**0.552**	0.131
AUC	0.622	0.777	0.673	0.802	0.721	0.717	**0.719**	0.141
From Scratch								
F1	0.402	0.533	0.460	0.509	0.438	0.555	0.483	0.143
AUC	0.578	0.754	0.589	0.685	0.652	0.701	0.660	0.140

with an autoregressive modeling loss, trying to predict the next token. For all experiments we used leave one out cross-validation over the patients.

Tables 5, 6 and 7 focus on the binary classification tasks. It is shown that the model performs better in some cases, such as the examples from when patients are drinking, shown in Table 6 while also performing worse, for example in the communication examples in Table 5.

This is also seen in Table 8 in which the task is to classify samples between Normal, PID or LID. The pretrained model is able to slightly outperform the model trained from scratch, but the variance is high and neither match the performance of Li et al. [2018b]. Table 9 is similar, with the pretrained model slightly outperforming the model trained from scratch, where the task is to infer a physician validated score of Parkinson's severity. These scores, UPDRS and UDysRS, are standardised clinical scores produced by human clinical annotation.

Table 7: Performance on binary classification of leg agility examples.

	Rleg	Lleg	Mean	Sigma
Li et al. [2018b] F1	0.538	0.735	**0.631**	
Li et al. [2018b] AUC	0.699	0.842	**0.770**	
Pretrained				
F1	0.436	0.470	0.453	0.081
AUC	0.542	0.547	0.545	0.103
From Scratch				
F1	0.422	0.415	0.419	0.066
AUC	0.700	0.616	0.658	0.187

Table 8: Performance on multiclass classification.

	Accuracy	Sigma
Li et al. [2018b]	**71.4%**	
Pretrained	68.3%	14.1%
From Scratch	64.3%	12.5%

Table 9: Performance on UPDRS and UDysRS score regression.

	UDysRS Part III	UPDRS Part III
Li et al. [2018b] RMS	**2.906**	**7.765**
Pretrained	3.232 +/- 1.206	10.586 +/- 3.778
From Scratch	3.447 +/- 1.284	10.609 +/- 4.036