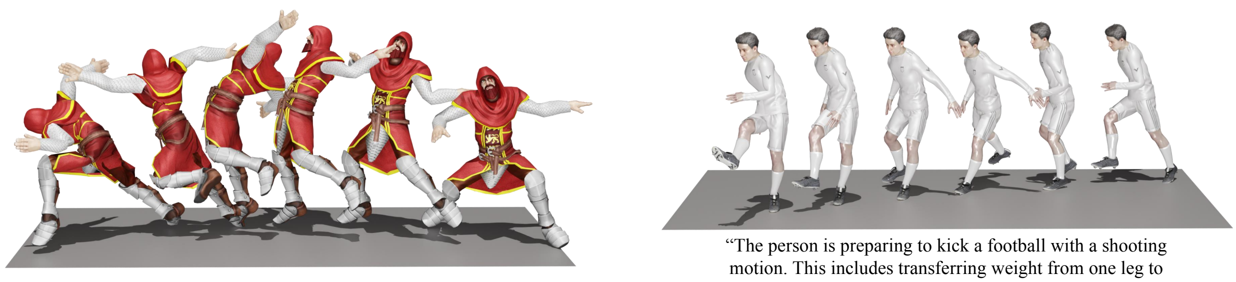
# ScaMo: Exploring the Scaling Law in Autoregressive Motion Generation Model

Shunlin Lu[123], Jingbo Wang[3]†, Zeyu Lu[5], Ling-Hao Chen[4], Wenxun Dai[4],
Junting Dong[3], Zhiyang Dou[6], Bo Dai[36], Ruimao Zhang[1]

[1] Sun Yat-sen University [2] The Chinese University of Hongkong, Shenzhen
[3] Shanghai AI Laboratory [4] Tsinghua University
[5] Shanghai Jiao Tong University [6] The University of Hong Kong



"Heavy attack forward"

"The person is preparing to kick a football with a shooting motion. This includes transferring weight from one leg to the other, swinging one leg back and then forward in a kicking gesture, while the arms balance the body."

Figure 1. The generation results of ScaMo-3B with a text input. Our model could deal with abstract sentences and long sentences.

## Abstract

*The scaling law has been validated in various domains, such as natural language processing (NLP) and massive computer vision tasks; however, its application to motion generation remains largely unexplored. In this paper, we introduce a scalable motion generation framework that includes the motion tokenizer Motion FSQ-VAE and a text-prefix autoregressive transformer. Through comprehensive experiments, we observe the scaling behavior of this system. For the first time, we confirm the existence of scaling laws within the context of motion generation. Specifically, our results demonstrate that the normalized test loss of our prefix autoregressive models adheres to a logarithmic law in relation to compute budgets. Furthermore, we also confirm the power law between Non-Vocabulary Parameters, Vocabulary Parameters, and Data Tokens with respect to compute budgets respectively. Leveraging the scaling law, we predict the optimal transformer size, vocabulary size, and data requirements for a compute budget of* $1e18$*. The test loss of the system, when trained with the optimal model size, vocabulary size, and required data, aligns precisely with the predicted test loss, thereby validating the scaling law.*

† Corresponding author.

## 1. Introduction

Scaling law gives the precise prediction of test loss, optimal model size, and data requirements given a specific compute budget in FLOPs. This capability enables researchers to conduct relatively small-scale experiments and accurately forecast the performance on larger scales, thereby conserving research time and compute resources. In recent years, the scaling law has been extensively studied, particularly in the field of natural language processing (NLP), where many large language models (LLMs) [1, 5, 9, 28] have empirically validated their scaling properties.

Building on these prior successes, recent research has extended scaling laws to the community of computer vision, particularly for tasks such as text-to-image generation [31, 50]. However, the scaling properties within the realm of human motion generation remain relatively underexplored. This is largely due to the challenges associated with the costly process of motion data collection and the substantial computational resources required. To guide the allocation of data collection efforts and compute budgets for further training, our objective is to examine whether similar scaling behaviors can be observed in motion generation tasks. Specifically, we take a transformer-based decoder-only auto-regressive motion generation framework as an example. To reach a targeted test loss, we need to set the computation budget accordingly, which can further be used to determine the optimal data requirements, appropriate model

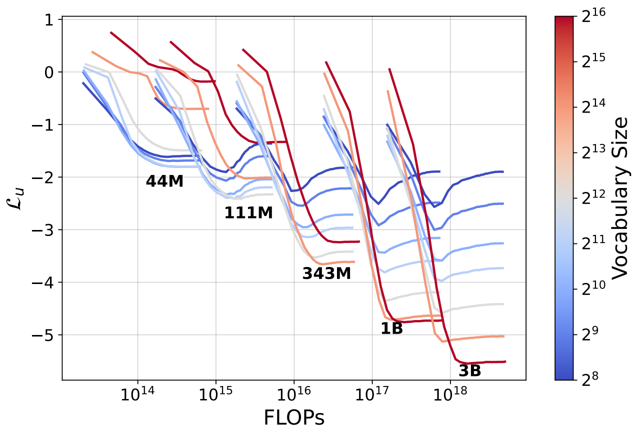



Figure 2. We plot the relationship between normalized test loss and FLOPs for observing the scaling behavior. Overall, the larger model and larger vocabulary size can get better performances.

parameters, and vocabulary size.

Exploring the scaling law of the auto-regressive model is natural and applicable in text-driven motion generation. However, building a motion generation framework at a larger scale has still not been explored well in the community. In this work, we mainly try to answer a research question, *What hinders the verification of scaling law in text-driven motion generation?* To this end, we attempt to answer this question from the following aspects.

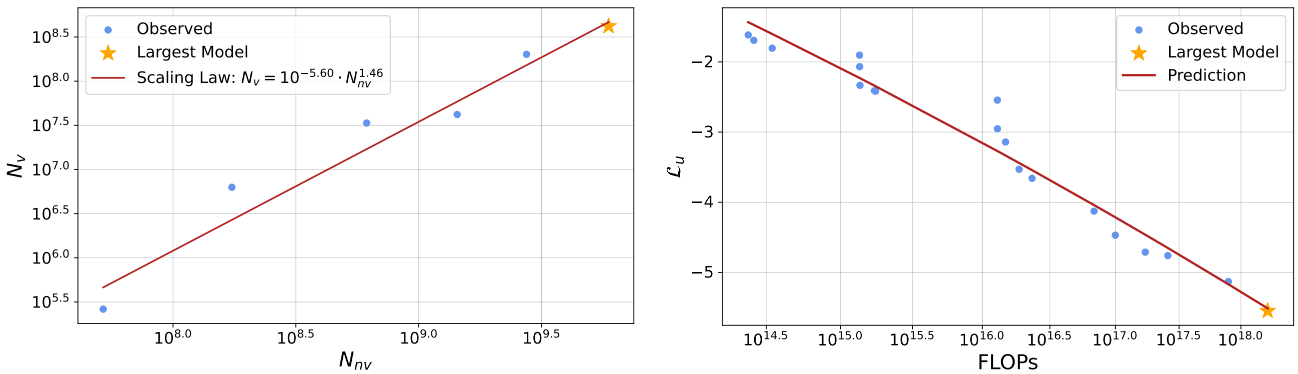
(i) **Limited data scale and quality.** The scale of motion data is much less than languages, images, or videos, due to its expensive collection process. Specifically, the largest used dataset Motion-X [32] only contains 98,000 sequences, which is not enough for observing the scaling properties. A concurrent work [58] collected a dataset of over 1M motion sequences. However, in this dataset, over half of the sequences consist of one-frame pose repeated 64 times, introducing a significant number of static motions.

(ii) **Difficulties in scaling the vocabulary size of tokenizer.** In the auto-regressive motion generation framework, directly scaling the size of the motion codebook is almost in vain. In contrast to tokenizers used in text, image, and video modalities, the vocabulary in motion tokenization is insufficient for naïve scaling. Directly applying existing vector quantization (VQ) methods for tokenization fails to scale effectively. When the codebook size increases, VQ suffers from codebook collapse, resulting in low utilization of codebooks. Therefore, exploring an effective tokenizer for auto-regressive human motion generation is urgent.

(iii) **Insufficient scalability of the model architecture.** Previous work tries to introduce a large language model with an extended vocabulary [27, 58], which compromises the generation performance with diverse downstream tasks, such as motion understanding. Furthermore, directly introducing the LLMs into motion generation makes it hard to explore the scalability of model size, due to the limited size choices of pretrained foundational LLMs. Other works train autoregressive models from scratch [67]. They use the sentence embedding from CLIP [43]. However, recent work [12] points out that the sentence-level language guidance for motion generation is not fine-grained enough for cross-modality alignment. Therefore, how to marriage foundational text encoders with a large motion generation model is still not well studied in the community.

To resolve the limitations above, we present a scalable system, denoted as ScaMo, to better investigate the scaling properties in text-driven motion generation. Unlike previous works directly using the existing dataset, we first collect a dataset of over 260 hours of motion data from sources such as Motion-X [32], CombatMotion [65], 100-Style [37], and an internal dataset. This dataset, denoted as MotionUnion, does not suffer from the static motion issue that was observed in previous studies [58]. For the motion tokenization process, we adopt a more generalizable approach, finite scale quantization (FSQ) [38], only relying on a simple reconstruction loss. This method can maintain performance similar to Vanilla VQ [67] at a relatively small codebook size. In contrast, when scaling with a larger codebook size, FSQ mitigates the codebook collapse issue and achieves superior performance. Additionally, to avoid compromising the language compression capability of the model, we use a frozen T5-XL [44] as the encoder. Importantly, the text is encoded as word-level embeddings prefixed in the auto-regressive motion generation model. In this transformer-based motion generator, we apply bidirectional attention to text tokens and causal attention to motion tokens. The text tokens are visible to all motion tokens. These improvements enable us to perform a comprehensive study on the scalability of motion generation models. We plot the relationship between test normalized loss and FLOPs in Fig. 2. Especially, we observe the scaling behaviors for the first time in motion generation.

Our experiments reveal several important insights: **Logarithmic law relationship between normalized test loss and FLOPs**. We observe a logarithmic relationship between the normalized test loss and computational resources (FLOPs). From this, we can predict the achievable test loss for a given FLOPs. In the absence of computational constraints, larger FLOPs should be preferred to maximize performance. This implies that larger vocabulary sizes and larger transformers should be selected. The larger vocabularies make the model more expressive, and a larger vocabulary requires a larger transformer. **Power law between vocabulary size, model size, and data token with respect to FLOPs.** With the given FLOPs, we can predict the optimal vocabulary size, model size and data tokens to get the best performance. Furthermore, we find the power relationship between non-vocabulary parameters and vocabulary parameters. Our prediction also suggests that vocabulary param-




(a) Non-Vocabulary Parameter Scaling Law.

(b) Performance Scaling Law.

Figure 3. Scaling laws of ScaMo. (a) Power law between $N_{nv}$ and $N_v$. We could predict the $Nv$ precisely based on a given $N_{nv}$. (b) Logarithmic law between FLOPs $C$ and normalized test loss $\mathcal{L}_u$. We could predict the $\mathcal{L}_u$ precisely given a FLOPs $C$.

eters should be scaled faster than non-vocabuolary parameters, i.e., $N_v \propto N_{nv}^{\gamma}$, where $\gamma \approx 1.46 > 1$, shown in Fig. 3.

To verify the effectiveness of the estimated scaling law, we set a fixed computation budget to $1 \times 10^{18}$. With the obtained scaling law, we predict the optimal model size, optimal vocabulary size, and required data are set as $3B$, $2^{16}$, and $1 \times 10^{7.5}$ respectively. When training a model in this setting, we found that the test normalized loss precisely aligns with the predicted loss, as shown in Fig 3. When training on our dataset, the model enjoys smooth generalization capabilities for diverse text inputs and generates higher-quality motions, whose results are shown in Fig. 1. We believe our model shows significant potential for scaling, with the capacity to improve motion generation quality on larger text-motion datasets. Additionally, it exhibits the ability for out-of-distribution text inputs.

Our contribution can be summarized as (1) We first demonstrate the existence of scaling laws in the motion generation task from a practical perspective. (2) We have revealed the core factors that limit the scaling laws in motion generation of previous work are the lack of data and unscalable model architectures. Our proposed scalable motion generation system opens the door to scalable research in motion generation. (3) The trained models can deal with complex sentences and generate more vivid motions.

## 2. Related Work

**Text-aligned human motion generation** [2–4, 7, 8, 12–14, 18–20, 25–27, 29, 33–35, 39, 40, 42, 47, 52, 53, 57, 59–63, 66–71, 73] develop fast in recent years, owing to both the progresses in generative models [23, 48, 49, 56] and the scales of datasets [17, 32, 64]. For methods, the introduction of GPT-like [19, 27, 35, 67] motion generation method and the diffusion-based method [13, 15, 53, 69, 70, 73] boost the development of human motion generation in a large extend. For datasets, KIT [41] and HumanML3D [17] are two representative datasets supporting human motion

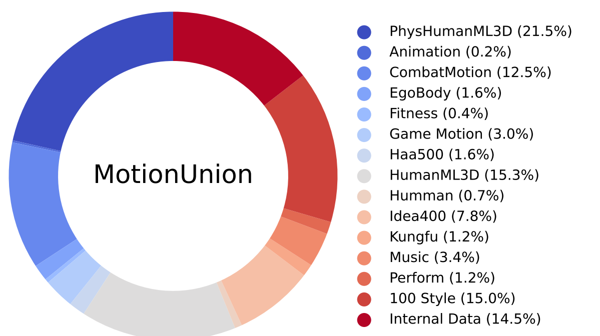



Figure 4. The frames statistics of MotionUnion dataset. Motion capture data accounts for the majority.

with open vocabulary inputs. Recent progress [30, 32, 64] in human motion generation tries to enhance the generation quality via scaling the size of datasets. However, how to synthesize human motion with more data, larger models and motion tokens is still under-explored.

**The scaling law in generative models** has attracted significant attention in recent years mainly due to the significant progress in auto-regressive language generation [1, 11, 55]. Besides, the progress in the visual generation community [10, 45, 46, 54] also verifies the effectiveness of scaling data, model size, and computation. Exploring the relationship between these resources and the model performance is extremely essential to predict the experimental results when scaling on resource-cost scenarios. Early research [6] has verified the power scaling law with the data scale. In recent years, some research in the language community [21, 22, 28] additionally investigates the scaling law with the model and data. Hoffmann *et al*. [24] show how to scale with the data and the model size jointly. Recent study [54] also verifies the power scaling law of auto-regressive models in image generation with model size and computational cost. However, how to scale with data, model size, and vocabulary size in the auto-regressive (sequential) motion generation process is still under exploration.

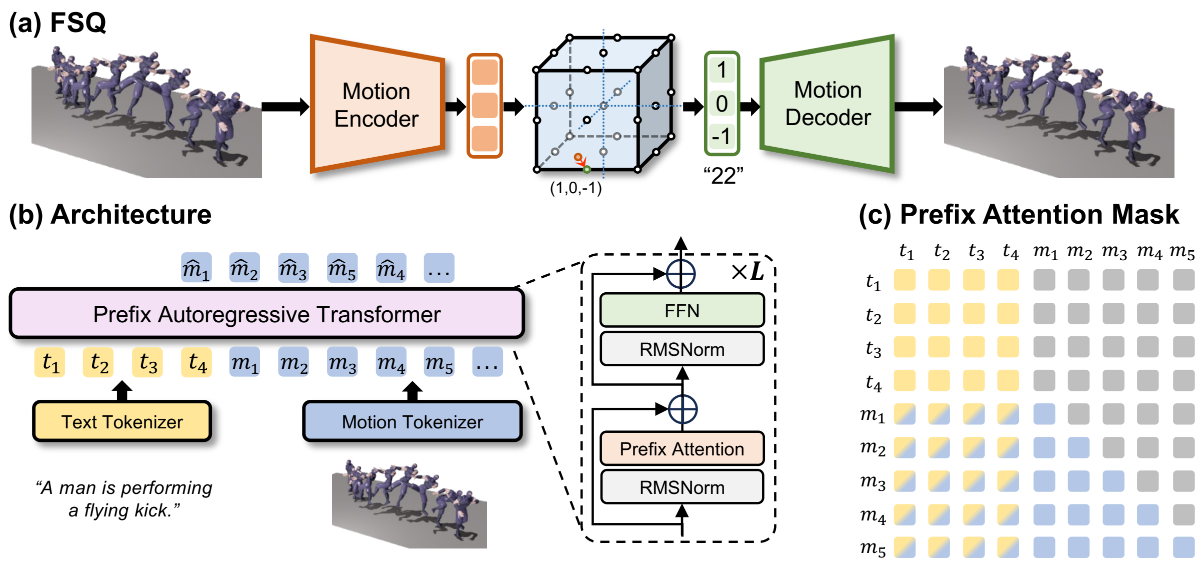



Figure 5. Overview of ScaMo architecture. **(a) FSQ**: Motion FSQ-VAE. We use one code quantization and $d = L = 3$ as an example. The feature of other frames is quantized in the same way. **(b) (c) Text-prefix Autoregressive Transformer**: The text tokens are applied with bidirectional attention and the motion tokens are applied with causal attention. Motion tokens can attend all text tokens.

## 3. Dataset Construction

Previous research [67] indicates that the existing dataset suffers from server overfitting, which limits its utility for evaluating scaling properties. To better explore these properties, we introduce a new, large-scale text-motion dataset, MotionUnion, comprising approximately 150k sequences and 30M frames. Detailed statistics are provided in the Appendix. Each sequence in the dataset is paired with a corresponding sentence. The dataset includes data from several sources, such as Motion-X [32], Combatmotion [65], 100 Style [65], and Physics-corrected HumanML3D, along with additional internal data. The internal data are sourced from manually created animations or motion capture, and are annotated with text generated by GPT-4. Following the methodology outlined in Smoodi [72], we retarget the raw motion data to the SMPL skeleton and employ the same processing pipeline as HumanML3D [17] to obtain the motion representations. Examples of the data can be found in the Appendix. The dataset composition and the proportion of each component are illustrated in Fig. 4.

## 4. Scalable Motion Generation System

Motion is inherently suitable for autoregressive modeling following tokenization. Similarly to previous work, our scalable framework comprises two key components: a motion tokenizer and a prefix autoregressive generation model, as depicted in Fig. 5. In the realms of image and video processing, the pursuit of large vocabulary sizes has become commonplace. To accurately capture motion details, a large vocabulary size is also essential the field of motion generation. However, the advantages of employing a large vocabulary size have not been widely recognized yet. Previous studies have not demonstrated notable improvements or robustness when utilizing large vocabularies. Upon further investigation, we hypothesize that this limitation arises from the codebook collapse phenomenon induced by the quantization method. To address this issue, we propose a scalable finite scalar quantizer, which is discussed in Sec. 4.1. Additionally, to enhance text encoding, we introduce the prefix autoregressive model in Sec. 4.2.

### 4.1. Finite Scalar Quantization

**Vanilla motion VQ-VAE.** Motion VQ-VAE is designed to learn discrete representations of human motion sequences through an encoding-decoding framework. Specifically, the model employs a vector quantization (VQ) mechanism to reconstruct motions, where an auto-encoder architecture is utilized. The model learns a codebook $\mathcal{C} = \{\mathbf{e}_k\}_{k=1}^{K}$, with $K$ representing the size of the codebook and $\mathbf{e}_k$ denoting the $k$-th embedding in the codebook.

Given a latent vector $\mathbf{z}$ and a quantizer $\mathcal{Q}(\cdot;\mathcal{C})$, the quantization process selects the codebook entry $\hat{\mathbf{z}}$ that minimizes the reconstruction error relative to $\mathbf{z}$, formally defined as,

$$\hat{\mathbf{z}} = \mathcal{Q}(\mathbf{z};\mathcal{C}) = \arg\min_{\mathbf{e}_k} \|\mathbf{z} - \mathbf{e}_k\|_2^2. \quad (1)$$

In the context of vanilla Motion VQ-VAE, the latent vector $\mathbf{z}$ is derived from the motion encoder $\texttt{Enc}(\cdot)$ applied to a motion sequence $\mathbf{m} \in \mathbb{R}^{T \times D}$, where $T$ is the number of motion frames and $D$ is the feature dimension of each frame. The motion is further reconstructed through the decoder $\texttt{Dec}(\cdot)$, with the overall model optimized to minimize the following loss function,

$$\mathcal{L} = \|\mathbf{m} - \texttt{Dec}(\mathcal{Q}(\mathbf{z};\mathcal{C}))\|_2^2 + \alpha\|\mathbf{z} - \texttt{sg}(\hat{\mathbf{z}})\|_2^2, \quad (2)$$

where $\alpha$ is a hyperparameter, $\texttt{sg}(\cdot)$ denotes the stop-gradient operation, and $\texttt{Dec}(\cdot)$ is the motion decoder.

Unlike traditional VQ-VAE frameworks, the codebook $\mathcal{C}$ in Motion VQ-VAE is updated using two techniques: exponential moving average (EMA) and codebook reset, as proposed in T2M-GPT [67]. While the use of discrete vector quantization in vanilla VQ-VAE effectively compresses human motion data, the quantization error remains significant. To address this issue, one might consider increasing the size of the codebook. However, a direct increase in codebook size can lead to codebook collapse, resulting in degraded performance and instability during training, especially as the model scales to handle more complex motion data, which is shown in Sec. 5.2.

**Motion FSQ-VAE.** To resolve the issue of codebook collapse, we analyze the key problem that lies in the "$\arg\min$" operation of vanilla VQ-VAE. The matching process is done by comparing the distance between $\mathbf{z}$ and $\hat{\mathbf{z}}$ via "$\arg\min$" operation, leading the optimizer to prefer the specific parts of the codebook and ignore updating others. To resolve this issue, we use a finite scalar quantizer (FSQ) instead. Technically, instead of using $\arg\min$, FSQ tries to round $\mathbf{z}$ as,

$$\hat{\mathbf{z}} = \mathcal{Q}(\mathbf{z}) = \texttt{round}(f(\mathbf{z})), \tag{3}$$

where $f(\cdot)$ is the bounding function, setting as the $\texttt{sigmoid}(\cdot)$ function in our practice. Each channel in $z$ will be quantized into one of the unique $L$ integers, therefore we have $\hat{z} \in \{1, \ldots, L\}^d$. The codebook size is calculated as $|\mathcal{C}| = \prod_{i=1}^{d} L_i$. Here, $L$ is a super parameter, and we follow Mentzer *et al.* [38] to set $L_i \geq 5$. We leave the $L_i$ we use in the appendix. Similarly, the $\texttt{round}(\cdot)$ operation can not propagate gradients, thus the stop-gradient technique is used. The optimization objective is only the reconstruction loss is required without other tricks, *i.e.*,

$$\mathcal{L} = \|\mathbf{m} - \texttt{Dec}(f(z) + \texttt{sg}(\texttt{round}(f(z)) - f(z)))\|_2^2, \tag{4}$$

where the encoders and decoders are adapted from vanilla VQ-VAE [67]. Note that FSQ is a replacement for VQ and can also be extended to group FSQ or residual FSQ. We leave it as our further work.

### 4.2. Text Prefix Autoregressive Model

We revisit how the previous motion generator integrates with the foundational language models. We notice that directly extending the vocabulary of language to motion tokens is not effective enough for text-driven motion generation. Differently, we introduce the world-level language prefix for the generation process. In contrast to some classical auto-regressive motion generation models [67], the input sentence is encoded as tokens for each word. As illustrated in Fig. 5, the attention calculation within the word part is bidirectional while the attention calculation for the motion part is causal attention. In this way, we could leverage the text embedding from the frozen text encoder, and the prefixed auto-regressive model is only optimized by the motion tokens part using the cross-entropy loss,

$$\mathcal{L} = -\sum_{t=1}^{n} \log p(\hat{m}_t | m_{<t}, S, V), \tag{5}$$

where $S$ denotes the text and $V$ denotes the vocabulary.

### 4.3. Scaling Law Formulation in Our Model

Classical scaling law only supports the fixed vocabulary, thus we follow the previous work [51] to reformulate. With the given compute budget $C$ in FLOPs, the goal is to get the optimal non-vocabulary model parameters $N_{nv}$, the optimal vocabulary model parameters $N_v$, the number of training tokens $D$. $N_v$ is calculated as $N_v = Vd$, where $V$ is the vocabulary size and $d$ is the dimension of the model. Accordingly, these terms can be formulated as,

$$\begin{aligned} (N_v^{\text{opt}}, N_{nv}^{\text{opt}}, D^{\text{opt}}) &= \arg \min_{N_v, N_{nv}, D} \mathcal{L}(N_v, N_{nv}, D) \\ &\text{s.t.} \quad \text{FLOPs}(N_v, N_{nv}, D) = C, \end{aligned} \tag{6}$$

According to Kaplan *et al.* [28], the FLOPs ($C$) of our Transformer-based model can be estimated as,

$$C \approx 6ND = 6(N_v + N_{nv})D \approx 6(N_{nv} + Vd)D. \tag{7}$$

To capture the scaling law in text-driven motion generation, we follow Hoffmann *et al.* [24] to train models with different model sizes from 44M-3B and vocabulary sizes from $2^8$ to $2^{16}$. Since the model with a large vocab size naturally has a higher cross-entropy loss. We follow Tao *et al.* [51] to use the normalized loss for fair evaluation,

$$\mathcal{L}_u = -\frac{1}{T} \sum_{t=1}^{T} \log \frac{p(m_t | m_{<t}, S, V)}{p(m_t | S, V)}, \tag{8}$$

Then we plot the IsoFLOPs figure, shown in Fig. 2. Similarly to previous work, we hypothesize the power law equations can capture the relationship between quantities. We formulate the relationship as,

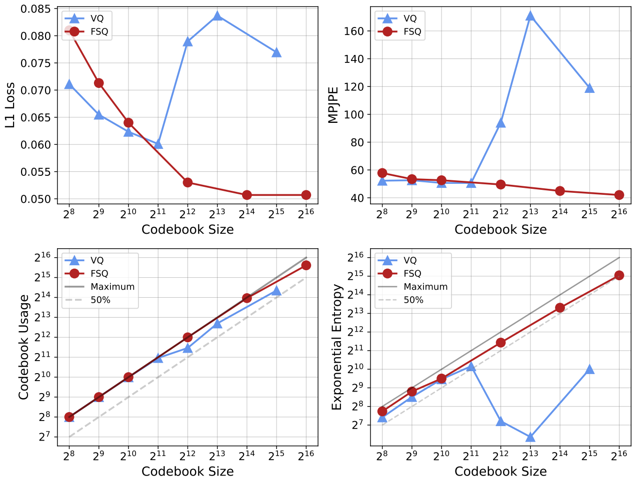
$$N_v^{\text{opt}} \propto C^a, \quad N_{nv}^{\text{opt}} \propto C^b, \quad \text{and} \quad D^{\text{opt}} \propto C^c. \tag{9}$$

We choose the pre-defined FLOPs that can reach the lowest normalized loss and fit the above power law equations to obtain the above coefficients.

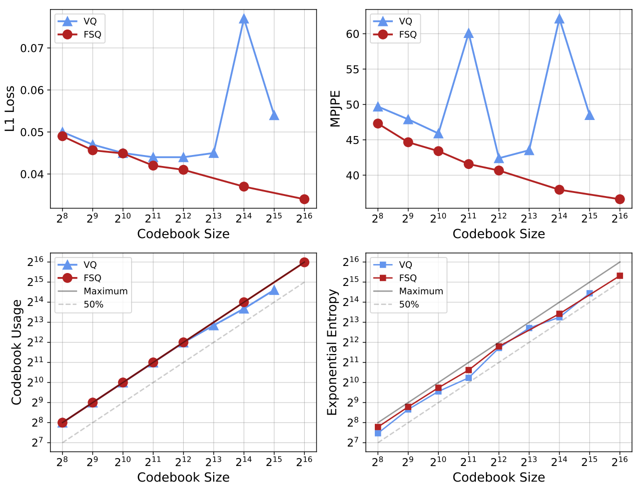
## 5. Experiments

In our experiments, we aim to answer the following research questions and conduct corresponding ablation studies.

- **RQ1:** Is FSQ-VAE more effective and scalable than the vanilla VQ-VAE?
- **RQ2:** What model and vocabulary sizes should we use to get the best results with infinite computation resources?
- **RQ3:** Given a pre-defined compute budget in FLOPs, how should we choose the model size, the vocabulary parameters and how much data should we collect?
- **RQ4:** Can we predict the normalized loss based on a given computation budget, like FLOPs?




(a) HumanML3D.




(b) MotionUnion.

Figure 6. Reconstuction results of different tokenizers on HumanML3D and MotionUnion. Reconstruction: L1 loss and MPJPE. Codebook Utilization: Codebook Usage and Entropy.

| Parameters | Layers | Heads | Hidden size |
|---|---|---|---|
| ScaMo-44M | 8 | 8 | 512 |
| ScaMo-111M | 12 | 12 | 768 |
| ScaMo-343M | 24 | 16 | 1024 |
| ScaMo-775M | 36 | 20 | 1280 |
| ScaMo-1.4B | 48 | 24 | 1536 |
| ScaMo-3B | 24 | 32 | 3200 |

Table 1. Model Configuration Table

## 5.1. Experiments Settings

**Evaluation dataset**. We evaluate the motion tokenizer using HumanML3D and our MotionUnion. HumanML3D dataset contains 14,616 motions from AMASS [36] and HumanAct12 [16], and with along 44,970 descriptions. The evaluation part of MotionUnion is a held-out dataset from the original whole dataset and is divided into training, testing, and validation with a ratio of $80\% : 15\% : 5\%$.

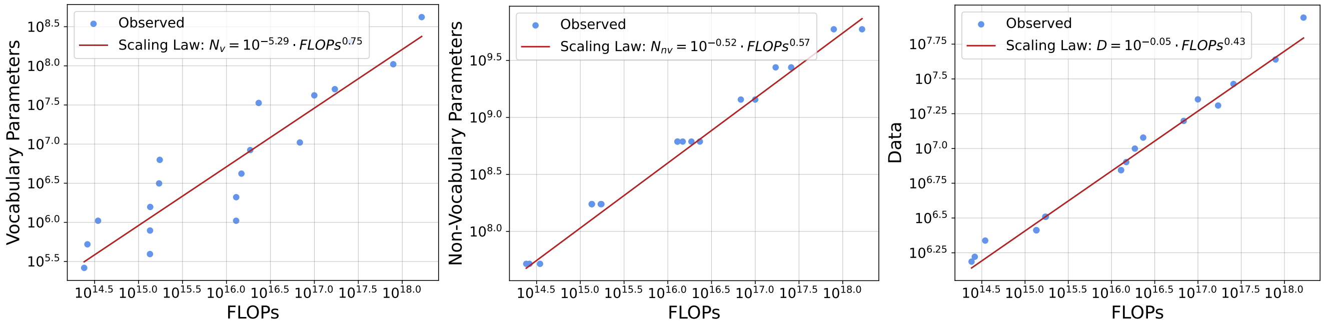
**Implementation details.** For the Motion FSQ-VAE, both the encoders and decoders are designed as convolutional residual blocks, utilizing a downsampling factor of 4. Regarding the prefix motion transformers, detailed specifications and configurations are provided in Tab. 1, where layers indicate the number of transformer decoder blocks. The transformer architecture closely aligns with that of LLaMA. Specifically, each block incorporates RMSNorm prior to both the prefix attention layer and the feed-forward network (FFN) layer. Due to space constraints, additional implementation details are presented in the appendix.

**Evaluation Metrics.** We use reconstruction loss, mean per joint position error (*a.k.a.* MPJPE), codebook utilization, and exponential entropy (*a.k.a.* Entropy) to evaluate motion tokenizers. The reconstruction loss and MPJPE reflect the reconstruction performance, while the codebook utilization and Entropy reflect the percentage of codes used in the test set. What we are pursuing is the lower reconstruction loss and MPJPE. The ideal motion tokenizer should have almost full codebook utilization and a higher Entropy. For the generation results, we conduct experiments on the HumanML3D benchmark. Following previous work, we use FID, R-precision, and matching score to evaluate the generation results. For validation of scaling law, we found FID is biased to the motion domain, and the pretrained motion feature extractor makes it hard to distinguish the differences between different models. Therefore, we follow previous work [51] to use the normalized loss described in Eq. (8).

## 5.2. Experiments on Different Tokenizers (RQ1)

We present comprehensive experimental results across a range of codebook sizes, which are illustrated in Fig. 6. Numerical values are provided in the appendix. As shown in Fig. 6, the results clearly indicate that FSQ consistently outperforms VQ across various metrics on both the smaller HumanML3D dataset and the larger MotionUnion dataset. Notably, FSQ achieves comparable reconstruction performance to VQ with smaller codebook sizes. Additionally, as the codebook size increases, it performs much better on both reconstruction Loss and MPJPE than VQ, which suggests that FSQ can reconstruct data more accurately, particularly as the vocabulary size expands. This key observation demonstrates a clear advantage over VQ in terms of reconstruction fidelity. The benefits of FSQ become especially evident at larger codebook sizes, where it continues to maintain consistently lower error rates.

Moreover, FSQ exhibits the capacity to scale with larger codebook sizes while maintaining steadily increasing performance across datasets. As evidenced by the results in Fig. 6, the reconstruction Loss and MPJPE in FSQ steadily decrease with increasing codebook size, reflecting its robustness and stability when handling larger vocabularies. In contrast, the vanilla VQ exhibits instability, where




(a) Vocabulary Parameter Scaling Law. (b) Non-Vocabulary Parameter Scaling Law. (c) Data Scaling Law.

Figure 7. Power laws of vocabulary parameters, non-vocabulary parameters, and data with respect to FLOPs.

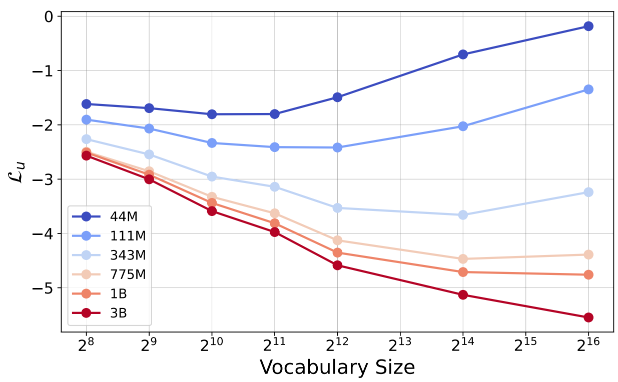



Figure 8. $\mathcal{L}_u$ curves under various model and vocabulary sizes.

its performance fluctuates significantly as the codebook size increases. This suggests that VQ struggles to achieve consistent performance when scaling to larger codebooks.

Additionally, FSQ demonstrates superior codebook utilization, even at very large codebook sizes. As can be seen in Fig. 6, the utilization of FSQ remains close to the theoretical maximum, indicating that it effectively leverages the available capacity of the codebook. This effective utilization stands in stark contrast to VQ, whose utilization rate decreases as the codebook size grows, thereby highlighting FSQ's greater efficiency in handling larger codebooks.

Furthermore, FSQ achieves higher Entropy, indicating a more uniform distribution of its codes. This uniformity is crucial as it suggests that FSQ mitigates risks such as code collapse, which occurs when certain codes dominate the encoding process. The consistently higher Entropy values observed for FSQ demonstrate its ability to utilize codes evenly across the codebook, even as the codebook size expands. In comparison, VQ shows significant fluctuations in Entropy, indicating a less stable and uniform distribution.

In summary, the superior performance of FSQ in terms of reconstruction accuracy, codebook utilization, and code distribution uniformity positions it as a more robust and scalable alternative than VQ. This advantage is particularly beneficial in scenarios that require high-capacity encoding, such as large-scale motion data, where effective codebook utilization and precise reconstruction are paramount.

### 5.3. Larger Model deserves Larger Vocabulary Sizes (RQ2)

When we have an infinite computation budget, how should we choose the vocabulary and model specifications? We plot the relationship between normalized test loss, vocabulary size, and model size in Fig. 8. Accordingly, we have the following observations.

**Larger models consistently outperform smaller ones across all codebook sizes.** This trend highlights the inherent advantage of larger models in capturing complex patterns, as they achieve lower normalized test losses compared to smaller ones. The improved performance of larger models across the vocabulary range implies that their enhanced capacity allows them to represent language features with greater fidelity, even when the vocabulary size changes. This observation aligns with the general expectation in deep learning that scaling up model parameters.

**Increasing vocabulary size requires a correspondingly larger model to fully utilize the potential of the large codebook size.** Smaller models struggle to maintain low test loss as the vocabulary grows, However, larger models, particularly those with billions of parameters, continue to perform well or even improve. This suggests a synergistic relationship where larger models are capable of leveraging the diversity and complexity afforded by a more extensive vocabulary. As a result, they can generalize better across diverse motion token inputs, while smaller models appear constrained by limited capacity and therefore fail to benefit as much from the increased vocabulary size.

**Large model deserves large vocabulary sizes.** The data also indicate that larger models actually benefit from larger vocabularies, as shown by their improved performance up to a certain vocabulary threshold, beyond which the benefits may taper off or slightly diminish. Smaller vocabularies might restrict large models' representational power, curtailing their ability to perform optimally on complex language tasks. In essence, large models ”deserve” larger vocabularies, as this pairing allows them to reach their full potential.

**Power law between $N_v$ and $N_{nv}$.** Fig. 3 a demonstrates a strong power correlation between model parameters and vocabulary size, further underscores this relationship. The high $R^2$ value of 0.95 suggests that vocabulary size can be effectively predicted based on model size,

$$N_v = 10^{-5.604} \cdot N_{nv}^{1.467}. \tag{10}$$

We predefine 3B models with different vocabulary sizes. We draw the best $N_v$ in Fig. 3 a with a yellow star. The yellow star can align with the fitted line, further validating this power law. This law in Eq. (10) provides a practical tool to determine an optimal vocabulary size for a given transformer model size. We hope this power law can help the community choose the correct vocabulary size to get the best performance when using auto-regressive transformers.

We answer RQ2 here. When we have no computation budgets, we should use a larger model and vocabulary size, but these should still adhere to the power law above.

### 5.4. Power Law in Model Size, Vocabulary Size, and Data (RQ3)

As demonstrated in Sec. 4.3, we extend the methodology proposed by Chinchilla [24] by conducting a series of comprehensive experiments that explore variations in both model size and vocabulary size, as illustrated in Fig. 2. For each combination of model and vocabulary sizes, we identify the triplet $(N_{nv}, N_v, D)$ that minimizes the test loss. Subsequently, we visualize the relationship between model size, vocabulary size, and motion tokens with respect to FLOPs, as shown in Fig. 7. We hypothesize that these attributes follow a power-law relationship with FLOPs, expressed as $N_v^{\text{opt}} \propto C^a, \quad N_{nv}^{\text{opt}} \propto C^b, \quad D^{\text{opt}} \propto C^c$..

To fit this power-law model, we first determine $C^c$ based on the data, subject to the constraint $(C^a + C^b) \cdot C^c = C$. We then optimize the coefficients $a$ and $b$ corresponding to the optimal vocabulary size $N_v^{\text{opt}}$ and the optimal non-vocabulary size $N_{nv}^{\text{opt}}$. To empirically validate this hypothesis, we employ least squares estimation (LSE) to fit the model to the data.

**Power law between $N_v$, $N_{nv}$, and $D$ with $C$.** From our estimation, the optimal model parameters and tokens scale with the compute budget is as,

$$N_v = 10^{-5.29} \cdot C^{0.75}, \tag{11}$$

$$N_{nv} = 10^{-0.52} \cdot C^{0.57}, \tag{12}$$

$$D = 10^{-0.05} \cdot C^{0.43}, \tag{13}$$

where $N_{nv}/D \propto C^{1.325} > C$ indicates we should scale $N_nv$ faster than data. Similarly, $N_v/N_{nv} \propto C^{1.315} > C$ indicates we should scale $N_v$ faster than $N_{nv}$.

### 5.5. Scaling Law in FLOPs and The Test Loss (RQ4)

**Logarithmic law in FLOPs and normalized test loss.** Following the way of choosing representative points introduced in Sec. 5.4, we identify the optimal parameter combinations $(N_{nv}, N_v, D)$ that minimize test loss for a given FLOPs. The relationship between FLOPs and normalized test loss is depicted in Fig. 3 (b).

| Text Enc. | Prefix | FID ↓ | Matching Score ↑ | Top1 R-P ↑ |
|---|---|---|---|---|
| GT | - | - | 2.974 | 0.511 |
| CLIP | × | 0.226 | 3.422 | 0.402 |
| T5-XL | ✓ | **0.104** | **3.021** | **0.510** |

Table 2. Ablation experiments of the architecture.

We assume a logarithmic relationship between the computational cost $C$ and the normalized test loss $\mathcal{L}_u$, hypothesized as $\mathcal{L}_u \propto -\log_{10}(C)$. Same as Sec. 5.4, we employ least squares estimation (*a.k.a.* LSE) to fit a logarithmic curve to the data. The logarithmic scaling relationship can be expressed as,

$$\mathcal{L}_u = -1.062 \times \log_{10}(C) + 13.839, \tag{14}$$

which indicates that increasing the number of FLOPs consistently leads to a reduction in normalized test loss.

**Accurate prediction of normalized test loss using the logarithmic law.** To further validate the predictive accuracy of our proposed scaling law, we trained a large-scale model containing 3B parameters and a codebook size of $2^{16}$, resulting in a computational cost exceeding $10^{18}$ FLOPs. As shown in Fig. 7 (a), the fitted logarithmic curve closely aligns with the observed test loss values, thereby substantiating the robustness and generalizability of our scaling law for predicting model performance.

### 5.6. Ablation Studies

We conduct an ablation study on a scalable architecture by training two distinct model variants, both with a parameter size of 343M, consistent with the T2M-GPT configuration [67]. The first variant employs a CLIP text encoder without the text-prefix design, while the other utilizes a T5-XL text encoder with text-prefixed design. Both models are trained on our constructed dataset MotionUnion and evaluated on the HumanML3D benchmark. As shown in Tab. 2, the model with T5-XL and prefix design achieves a substantial performance improvement, with the FID decreasing from 0.226 to 0.104, the matching score increasing from 3.422 to 3.021, and the Top-1 R-Precision improving from 0.402 to 0.510. These results underscore the effectiveness of the text-prefixed autoregressive transformer architecture.

## 6. Conclusion

In this paper, we present a scalable text-driven motion generation system, comprising a motion FSQ-VAE and a text-prefix autoregressive transformer. Additionally, we provide the first empirical analysis of scaling behaviors within the motion generation domain. Our experiments reveal a logarithmic relationship between the compute budgets in FLOPs and normalized test loss. Furthermore, we observe that both vocabulary size and non-vocabulary size, as well as the amount of data, exhibit power-law dependencies with respect to FLOPs. We also identify a power-law relationship between vocabulary size and non-vocabulary size. We hope that our findings will offer valuable insights for future research and practical applications in the field of motion synthesis.

## References


[1] OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haim ing Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Benjamin Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Sim'on Posada Fishman, Juston Forte, Is abella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Raphael Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Lukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Ryan Kiros, Matthew Knight, Daniel Kokotajlo, Lukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Ma teusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel P. Mossing, Tong Mu, Mira Murati, Oleg Murk, David M'ely, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Ouyang Long, Cullen O'Keefe, Jakub W. Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alexandre Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Pondé de Oliveira Pinto, Michael Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack W. Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario D. Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin D. Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas A. Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cer'on Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll L. Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qim ing Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report. 2023. 1, 3

[2] Hyemin Ahn, Timothy Ha, Yunho Choi, Hwiyeon Yoo, and Songhwai Oh. Text2action: Generative adversarial synthesis from language to action. In *ICRA*, pages 5915–5920, 2018. 3

[3] Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. In *3DV*, pages 719–728, 2019.

[4] Nikos Athanasiou, Mathis Petrovich, Michael J Black, and Gül Varol. Teach: Temporal action composition for 3d humans. In *3DV*, pages 414–423, 2022. 3

[5] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 1

[6] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pages 26–33, 2001. 3

[7] German Barquero, Sergio Escalera, and Cristina Palmero. Seamless human motion composition with blended positional encodings. In *CVPR*, pages 457–469, 2024. 3

[8] Uttaran Bhattacharya, Nicholas Rewkowski, Abhishek Banerjee, Pooja Guhan, Aniket Bera, and Dinesh Manocha. Text2gestures: A transformer-based network for generating emotive body gestures for virtual agents. In *VR*, pages 1–10, 2021. 3

[9] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong,

Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024. 1

[10] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 3

[11] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Ma teusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. 3

[12] Ling-Hao Chen, Wenxun Dai, Xuan Ju, Shunlin Lu, and Lei Zhang. Motionclr: Motion generation and training-free editing via understanding attention mechanisms. *arXiv preprint arXiv:2410.18977*, 2024. 2, 3

[13] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *CVPR*, pages 18000–18010, 2023. 3

[14] Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. Mofusion: A framework for denoising-diffusion-based motion synthesis. In *CVPR*, pages 9760–9770, 2023. 3

[15] Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. *ECCV*, 2024. 3

[16] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *ACM MM*, pages 2021–2029, 2020. 6

[17] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *CVPR*, pages 5152–5161, 2022. 3, 4

[18] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *ECCV*, pages 580–597, 2022. 3

[19] Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *CVPR*, pages 1900–1910, 2024. 3

[20] Bo Han, Hao Peng, Minjing Dong, Yi Ren, Yixuan Shen, and Chang Xu. Amd: Autoregressive motion diffusion. In *AAAI*, pages 2022–2030, 2024. 3

[21] 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. *arXiv preprint arXiv:1712.00409*, 2017. 3

[22] Joel Hestness, Newsha Ardalani, and Gregory Diamos. Beyond human-level accuracy: Computational challenges in deep learning. In *Proceedings of the 24th symposium on principles and practice of parallel programming*, pages 1–14, 2019. 3

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, pages 6840–6851, 2020. 3

[24] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 30016–30030, 2022. 3, 5, 8

[25] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *ACM SIGGRAPH*, 2022. 3

[26] Yiheng Huang, Hui Yang, Chuanchen Luo, Yuxi Wang, Shibiao Xu, Zhaoxiang Zhang, Man Zhang, and Junran Peng. Stablemofusion: Towards robust and efficient diffusion-based motion generation framework. *ACM MM*, 2024.

[27] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *NeurIPS*, 2024. 2, 3

[28] 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. *arXiv preprint arXiv:2001.08361*, 2020. 1, 3, 5

[29] Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *CVPR*, pages 2151–2162, 2023. 3

[30] Han Liang, Jiacheng Bao, Ruichi Zhang, Sihan Ren, Yuecheng Xu, Sibei Yang, Xin Chen, Jingyi Yu, and Lan Xu. Omg: Towards open-vocabulary motion generation via mixture of controllers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 482–493, 2024. 3

[31] Zhengyang Liang, Hao He, Ceyuan Yang, and Bo Dai. Scaling laws for diffusion transformers. *arXiv preprint arXiv:2410.08184*, 2024. 1

[32] Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-x: A large-scale 3d expressive whole-body human motion dataset. *NeurIPS*, 2024. 2, 3, 4

[33] Xiao Lin and Mohamed R Amer. Human motion modeling using dvgans. *arXiv preprint arXiv:1804.10652*, 2018. 3

[34] Jinpeng Liu, Wenxun Dai, Chunyu Wang, Yiji Cheng, Yansong Tang, and Xin Tong. Plan, posture and go: Towards open-world text-to-motion generation. *ECCV*, 2024.

[35] Shunlin Lu, Ling-Hao Chen, Ailing Zeng, Jing Lin, Ruimao Zhang, Lei Zhang, and Heung-Yeung Shum. Humantomato: Text-aligned whole-body motion generation. *ICML*, 2024. 3

[36] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *ICCV*, pages 5442–5451, 2019. 6

[37] Ian Mason, Sebastian Starke, and Taku Komura. Real-time style modelling of human locomotion via feature-wise transformations and local motion phases. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 5(1):1–18, 2022. 2

[38] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023. 2, 5

[39] Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In *ECCV*, pages 480–497, 2022. 3

[40] Mathis Petrovich, Or Litany, Umar Iqbal, Michael J Black, Gul Varol, Xue Bin Peng, and Davis Rempe. Multi-track timeline control for text-driven 3d human motion generation. In *CVPRW*, pages 1911–1921, 2024. 3

[41] Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big data*, 4(4):236–252, 2016. 3

[42] Matthias Plappert, Christian Mandery, and Tamim Asfour. Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks. *RAS*, 109:13–26, 2018. 3

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[44] 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. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 2

[45] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 3

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3

[47] Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H Bermano. Human motion diffusion as a generative prior. In *ICLR*, 2024. 3

[48] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265, 2015. 3

[49] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 3

[50] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 1

[51] Chaofan Tao, Qian Liu, Longxu Dou, Niklas Muennighoff, Zhongwei Wan, Ping Luo, Min Lin, and Ngai Wong. Scaling laws with vocabulary: Larger models deserve larger vocabularies. *arXiv preprint arXiv:2407.13623*, 2024. 5, 6

[52] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *ECCV*, pages 358–374, 2022. 3

[53] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. In *ICLR*, 2022. 3

[54] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024. 3

[55] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 3

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 3

[57] Weilin Wan, Zhiyang Dou, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. Tlcontrol: Trajectory and language control for human motion synthesis. *ECCV*, 2024. 3

[58] Ye Wang, Sipeng Zheng, Bin Cao, Qianshan Wei, Qin Jin, and Zongqing Lu. Quo vadis, motion generation? from large language models to large motion models. *arXiv preprint arXiv:2410.03311*, 2024. 2

[59] Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. Humanise: Language-conditioned human motion generation in 3d scenes. *NeurIPS*, pages 14959–14971, 2022. 3

[60] Zan Wang, Yixin Chen, Baoxiong Jia, Puhao Li, Jinlu Zhang, Jingze Zhang, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. Move as you say interact as you can: Language-guided human motion generation with scene affordance. In *CVPR*, pages 433–444, 2024.

[61] Zeqi Xiao, Tai Wang, Jingbo Wang, Jinkun Cao, Wenwei Zhang, Bo Dai, Dahua Lin, and Jiangmiao Pang. Unified human-scene interaction via prompted chain-of-contacts. In *ICLR*, 2024.

[62] Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. Omnicontrol: Control any joint at any time for human motion generation. In *ICLR*, 2024.

[63] Zhenyu Xie, Yang Wu, Xuehao Gao, Zhongqian Sun, Wei Yang, and Xiaodan Liang. Towards detailed text-to-motion synthesis via basic-to-advanced hierarchical diffusion model. In *AAAI*, pages 6252–6260, 2024. 3

[64] Liang Xu, Xintao Lv, Yichao Yan, Xin Jin, Shuwen Wu, Congsheng Xu, Yifan Liu, Yizhou Zhou, Fengyun Rao, Xingdong Sheng, et al. Inter-x: Towards versatile human-human interaction analysis. In *CVPR*, pages 22260–22271, 2024. 3

[65] Liao Yihao, Fu Yiyu, Cheng Ziming, and Wang Jiangfeiyang. Animationgpt:an aigc tool for generating game combat motion assets. https://github.com/fyyakaxyy/AnimationGPT, 2024. 2, 4

[66] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *ICCV*, pages 16010–16021, 2023. 3

[67] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *CVPR*, pages 14730–14740, 2023. 2, 3, 4, 5, 8

[68] Jiaxu Zhang, Xin Chen, Gang Yu, and Zhigang Tu. Generative motion stylization of cross-structure characters within canonical motion space. In *ACM MM*, 2024.

[69] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *ICCV*, 2023. 3

[70] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE TPAMI*, 2024. 3

[71] 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. In *AAAI*, pages 7368–7376, 2024. 3

[72] Lei Zhong, Yiming Xie, Varun Jampani, Deqing Sun, and Huaizu Jiang. Smoodi: Stylized motion diffusion model. *ECCV*, 2024. 4

[73] Wenyang Zhou, Zhiyang Dou, Zeyu Cao, Zhouyingcheng Liao, Jingbo Wang, Wenjia Wang, Yuan Liu, Taku Komura, Wenping Wang, and Lingjie Liu. Emdm: Efficient motion diffusion model for fast, high-quality motion generation. *ECCV*, 2024. 3