The Thesis Committee for Yili Wang
certifies that this is the approved version of the following thesis:

Directly Optimizing Evaluation Metrics to Improve Text to Motion

SUPERVISING COMMITTEE:

Raymond Mooney, Supervisor

Qixing Huang

Directly Optimizing Evaluation Metrics to Improve Text to Motion

by

Yili Wang

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science

The University of Texas at Austin
May 2023

Acknowledgments

I want to express my deepest gratitude to my advisor Prof. Raymond Mooney for his patience and guidance throughout my study. His spirit of innovation always motivates me to continue my research. I also would like to thank Jordan Voas who is the research partner and also a friend of mine for his generous help. In addition, I am grateful to Prof. Qixing Huang for his constant support of the project. Besides, I want to extend my appreciation to everyone who provides me with help along the way. Finally, I want to express from the bottom of my heart my gratitude to my family for their unconditional love and consistent support. I will always treasure this wonderful journey.

Abstract

Directly Optimizing Evaluation Metrics to Improve Text to Motion

Yili Wang, MS
The University of Texas at Austin, 2023

SUPERVISOR: Raymond Mooney

There is a long-existing discrepancy between training and testing process of most generative models including both text-to-text models like machine translation (MT), and multi-modal models like image captioning and text-to-motion generation. These models are usually trained to optimize a specific objective like log-likelihood (MLE) in the Seq2Seq models or the KL-divergence in the variational autoencoder (VAE) models. However, they are tested using different evaluation metrics such as the BLEU score and Fréchet Inception Distance (FID). Our paper aims to address such discrepancy in text-to-motion generation models by developing algorithms to directly optimize the target metric during training time. We explore three major techniques: reinforcement learning, contrastive learning methods, and differentiable metrics that are originally applied to natural language processing fields and adapt them to the language-and-motion domain.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Text-to-motion generation task recently draws more and more attention together with the rapid development of other multi-modal generative models like text-to-image and text-to-video. The goal of text-to-motion is to take as input a textual prompt that describes a motion and output realistic enough motions that reflect the input description. Various deep learning methods have been applied to the text-to-motion task ranging from RNN-based Seq2Seq model Plappert et al. (2018); Lin et al. (2018), transformer Tevet et al. (2022a) to the latest diffusion models Tevet et al. (2022b); Zhang et al. (2022). However, fewer efforts focus on the evaluation of the generated motions. Our project tries to approach the text-to-motion task from the perspective of evaluation.

The evaluation has long been a challenge for the text-to-motion task. On the one hand, most of the metrics including the average positional error (APE) and Féchet Inception Distance (FID) Heusel et al. (2017) are inherited from other multi-modal generative tasks like text-to-image. It is still questionable whether they can effectively measure the quality of generated motions in terms of naturalness (whether the motion looks real) and faithfulness (whether the motion corresponds to the text prompt). In our recent work Voas et al. (2022), we analyze the effectiveness of different evaluation metrics for text-to-motion by measuring their correlation with human judgment in these two dimensions collected on the Amazon Mechanical Turk (AMT).

Our project extends previous work further to re-consider the training process based on the findings in the evaluation aspect. One of the biggest problems with many multi-modal generative models is that they often suffer from the discrepancy between training and testing. More specifically, these models are trained on objectives like log-likelihood probability (MLE) but tested on a different metric like FID. As a result, the training performance is not directly related to the final evaluation results. Moreover, many metrics are not differentiable with respect to the model parameters and we are unable to incorporate them directly into the training loss. Therefore, we see a large gap in this field and attempt to initiate efforts to address the discrepancy problem in our project.

1.1 Research Statement

In our project, our ultimate goal is to directly optimize the evaluation metrics during training to address the problem of discrepancy between training and testing of text-to-motion generation models. We adopt existing ideas from the text-to-text generation domain including contrastive learning Liu et al. (2022) and differentiable metric Wang et al. (2019) to our text-to-motion task to directly optimize the metrics we examined in Voas et al. (2022) during training. The motivation for this project is that *if we can optimize the metric that best correlates with human judgment during training, our model should achieve a better performance in terms of human evaluation.*

1.2 Contributions

Our main contributions are:

- A review of different machine learning methods in the text generation domain that aim to directly optimize evaluation metrics during the training process.
- The development of algorithms including contrastive learning approach and differentiable metrics approach by adapting techniques in the text generation domain to text-to-motion generation task.
- Thorough experiments and analysis to show the effectiveness of our proposed algorithms to improve text-to-motion generation performance by directly optimizing evaluation metrics during training.

Chapter 2: Related Work

In this section, we provide a background literature review of the three research areas that are the most related to our project: Text-to-motion Generation, Training methods and Evaluation metrics.

2.1 Text-to-Motion Generation

Text-to-motion as a specific multi-modal generation task develops rapidly with the progress in deep generative models. At the early stage, most models follow the sequence-to-sequence manner. Plappert et al. (2018) and Lin et al. (2018) applied RNN-based text encoder and motion decoder to generate pose distribution and raw joint values respectively. To mitigate the large gap between the modality of language and motion, JL2P Ahuja and Morency (2019) added an additional RNN-based motion encoder to train an autoencoder for motion modality and then map text features to learn a joint embedding space. Ghosh et al. (2021) extended the work in JL2P by applying hierarchically learn representations for different body parts and involving an additional motion discriminator for training which enable the GAN-like training process.

The downside of the Seq2Seq models is that they deterministically generate motions conditioned on input texts which means that texts and motions have a one-to-one correspondence that is unable to capture the diversity of motions in reality. More recent works moved to stochastic generative models like variational autoencoder (VAE) Petrovich et al. (2022); Guo et al. (2022a) which instead of directly generating the motion sequence, tried to model the latent distribution of motion and sample generation from it. Other attempts applied tokenized modeling Guo et al. (2022b) that used CNN to create a codebook of motion tokens and applied Seq2Seq model to generate the motion token sequence. Although this process is determinate, it involves diversity in the decoding process from motion tokens to the final motion sequence.

Furthermore, Diffusion models recently gain much attention because of their great success in the language-and-vision domain. This group of models learns to predict the noise at each time step of diffusing an image to the Gaussian noise so that the actual generation process is modeled as the denoising process which corresponds to the reversed diffusion. There are also some latest models that apply the diffusion method to the text-to-motion generation task, such as Tevet et al. (2022b); Zhang et al. (2022) which have achieved state-of-the-art performance.

2.2 Training Methods

Most Seq2Seq models in both text-to-text and multi-modal domains are trained with respect to the maximum likelihood estimation (MLE) which aims to optimize the conditional probability of the target word (motion) sequence. However, at testing time, the generated sequences are evaluated using completely different metrics such as BLEU score (text generation) Papineni et al. (2002) or FID Heusel et al. (2017).

There are more existing works to address the discrepancy in text-to-text generation tasks like machine translation and text summarization. Reinforcement learning (RL) methods are the most straightforward ones. Ranzato et al. (2016) was the earliest to model text generation tasks as an RL problem by treating the Seq2Seq model as the policy and BLEU scores as rewards. At each time step, the policy model outputs a word in the vocabulary as taking an action based on the current state which may involve the input condition, word sequence generated so far and current time step. There are also related works in the multi-modal domain like image captioning Qi and Peng (2018); Liu et al. (2017). However, there are two significant limitations of RL methods. As is pointed out in Wu et al. (2018); Choshen et al. (2020), the training process of RL methods is time-consuming and not data efficient, and the lack of enough training trials often leads to large variances of results. Secondly, these RL methods rely on the discrete nature of text generation task and none of them can be directly extended to generation in continuous domains such as image and motion.

Other works focus on aligning training loss with the target metrics in a metric-agnostic manner which means that the evaluation metrics can be included in the training objectives even if they are not necessarily differentiable. Minimum risk training (MRT) Shen et al. (2016) computes the expected loss over a number of candidates using online sampling methods to allow arbitrary non-differentiable loss functions. In recent work, BRIO Liu et al. (2022) adopts contrastive learning and ranking-based loss over a non-deterministic distribution of candidates so that the model can generate higher-scoring texts with higher probability. This contrastive learning method achieves good performance on the text summarization task.

Finally, there are a limited number of works on the differentiable version of evaluation metrics like DEBLEU Wang et al. (2019) but their effectiveness needs further investigation because the paper did not explicitly show the correlation between DEBLEU and BLEU and the experiments did not prove whether training with DEBLEU can directly improve BLEU score.

2.3 Evaluation Metrics for Text-to-motion Generation

2.3.1 Coordinate Error (CE) Metrics

The key idea of CE metrics is to take the sequence of 3D coordinates from generated motions and the ground truths and compute the distance as the evaluation of generation quality. Ghosh et al. (2021) proposed two categories of average positional error (APE) which are defined as the averaged Euclidean distance between candidates and references, and the average variance error (AVE) is defined as the mean Euclidean distance between the motion variance along the temporal dimension.

Voas et al. (2022) further extended the CE metrics to introduce more variants that take into consideration the difference between the root joint (Root), body joints (Joint) and all together (Pose) denoted by pose joints. In addition, coordinates beyond position were also explored including velocity (VEL) and acceleration

(ACCE) by computing the first and second-order derivatives of 3D joint coordinates. By permuting all categories of variants, there are a total of 18 CE metrics.

2.3.2 Féchet Inception Distance

Féchet Inception Distance (FID) is first proposed in Heusel et al. (2017) and is widely applied to the evaluation of image generations and is adopted by Guo et al. (2020) to the motion domain. FID measures the distance between the distributions of candidates and ground truth motion features extracted by the pre-trained Inception network Szegedy et al. (2016). For two feature distributions D_1, D_2, the FID is computed as:

$$FID = ||\mu_1 - \mu_2||^2 + tr(\Sigma_1 + \Sigma_2 - 2\sqrt{\Sigma_1 \Sigma_2}) \tag{2.1}$$

Where μ_1, μ_2 are the mean feature vectors and Σ_1, Σ_2 are the covariance matrices of the two distributions. tr is the matrix trace operation. Because FID measures the distance between motion features at the distribution level, it can only serve as a model-level naturalness metric.

2.3.3 Representation-based Metrics

Representation-based metrics rely on pre-trained encoders for the textual description and motion to map sequential data to vector representations in the embedding space.

Guo et al. (2022a) proposed two representation-based metrics: R-precision and MultiModal distance as complementary metrics to evaluate faithfulness. R-precision is defined by the average accuracy at top-1 to top-3 places where the ground truth textual description is correctly retrieved from 31 randomly selected texts using Euclidean distance between the generated motion features and text features. MultiModal distance is computed as the average Euclidean distance between features of the generated motion and the corresponding textual description.

Voas et al. (2022) proposed a novel representation-based metric called Nearest Neighbor Captioning (NNC) method. Firstly, the generated motion is mapped to the feature space using the motion encoder. Then, all the motions in the selected reference set are also mapped to the same embedding space. The nearest neighbors with the smallest distance from the generated motion are retrieved as candidates. Finally, the quality of the motion generation is measured by comparing the original texts with the corresponding texts of the neighbor motions.

2.4 Background Work

As our project can be regarded as the following work of Voas et al. (2022). In this section, we will review some important findings from the previous work which our project is mostly based on.

This paper collected generated motion candidates using four existing models: T2M Guo et al. (2022a), TM2T Guo et al. (2022b), MDM Tevet et al. (2022b) and MotionDiffuse Zhang et al. (2022) and conduct human study to score faithfulness and naturalness of the candidate motions together with the ground truths from HumanML3D dataset Guo et al. (2022a). Surprisingly, even though diffusion models start to dominate text-to-image and text-to-video areas, human evaluation reveals that diffusion models lack faithfulness and the encoder-decoder-based models can achieve better faithfulness while still presenting natural enough motions. Among these models, T2M model Guo et al. (2022a) receives the highest faithfulness scores and the second highest naturalness score which is only slightly less than MDM Tevet et al. (2022b).

Besides model evaluation, this paper also provides a thorough analysis of existing automated evaluation metrics for text-to-motion generation task ranging from Coordinate Error (CE) metrics which simply compute the distance between the 3D coordinates including joint positions, velocities and accelerations of generated motions and ground truths, representation-based metrics such as FID Heusel et al. (2017) and others that proposed in Guo et al. (2020) including multi-modal distance, R-precision and multi-modality. Although most of the current works prefer representation-based metrics over CE metrics possibly because they capture more sequence-level semantics, by computing the correlation with human judgment both at model level and sample level, the paper shows that the CE metrics can still outperform metrics like FID and R-precision. According to the paper, among different variants of CE metrics, the pose position error (Pose POS APE) has the highest Pearson's correlation with human scores at the sample level while the velocity position error (Pose VEL APE).

Finally, the paper proposes a novel automated text-to-motion evaluation metric called Nearest Neighbor Captioning (NNC) Method. Given the generated motion and the textual description, the NNC method relies on the pre-trained encoder to first map the motion to the embedding space and retrieves the nearest neighbor motion representations with the smallest Euclidean distance and also their corresponding from a reference dataset. The measurement of the generated motion quality can be transferred to the evaluation of their corresponding textual descriptions. There are several existing and mature tools in the text domain including BLEU score Papineni et al. (2002), BERTScore Zhang et al. (2019) and BLEURT Sellam et al. (2020).

Chapter 3: Text-to-Motion Generation

In this section, we will introduce and provide the general definitions for the important concepts and terminologies that we use throughout out the remaining sections. To keep consistency between different algorithms and simplicity of discussion, we also use the same set of mathematical symbols as defined in this section.

3.1 Problem Formulation

We give some general notations for the text-to-motion generation task. Given the textual description of totally L words $X = \{x_1, ..., x_L\}$ as input, where x_i is a word in the vocabulary, we want to train a text-to-motion generation model to generate 3D motion sequence $P = \{p_1, ..., p_T\}$ of total length T where $p_t \in \mathbb{R}^{J\times 3}$ is the 3D pose with totally J joints at time t.

3.2 Dataset

We use the HumanML3D dataset Guo et al. (2022a) which provides $14,616$ motions and $44,970$ textual descriptions. Each motion has at least three descriptions. To augment the data and improve motion diversity in the dataset, each motion has its mirrored version by flipping the keyword such as 'left' to 'right'. For the skeleton structure, the total number of joints is 22 and the pose is represented by the 3D coordinates $p \in \mathbb{R}^{J\times 3}$. Besides the skeleton representation, the HumanML3D dataset also defines a vector $(\dot{r}^a, \dot{r}^x, \dot{r}^z, r^y, \mathrm{j}^p, \mathrm{j}^v, \mathrm{j}^r, \mathrm{c}^f)$ of 263 dimensions to represent the pose at each timestep. The components include root angular velocity along Y-axis $\dot{r}^a \in \mathbb{R}$, root linear velocity on the XZ-plane $\dot{r}^x, \dot{r}^z \in \mathbb{R}$, root height $r^y \in \mathbb{R}$, and local joints positions $\mathrm{j}^p \in \mathbb{R}^{3j}$, velocities $\mathrm{j}^v \in \mathbb{R}^{3j}$, rotations in the root space $\mathrm{j}^r \in \mathbb{R}^{6j}$ where $j = J - 1 = 21$ is the total number of body joints, and the final binary features indicating foot ground contacts $\mathrm{c}^f \in \mathbb{R}^4$. The two representations are equivalent and can be transferred to each other using rotational and transitional matrices which are implemented in Guo et al. (2022a). In our project, we also adopt the same train-val-test split of the dataset. There are $23,384$, $1,460$ and $4,382$ distinct motions including the mirrored ones in the training, validation and testing sets respectively. Figure 3.1 presents the example motions from the HumanML3D dataset.

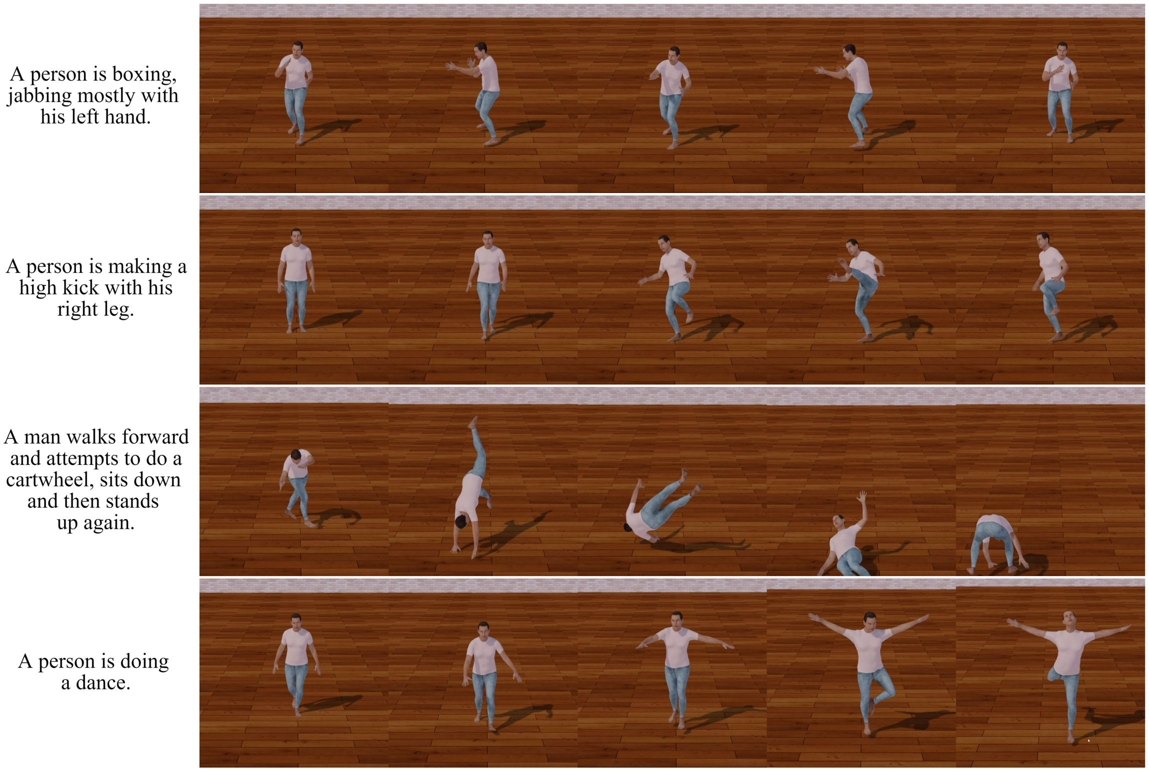

A person is boxing, jabbing mostly with his left hand.
A person is making a high kick with his right leg.
A man walks forward and attempts to do a cartwheel, sits down and then stands up again.
A person is doing a dance.

Figure 3.1: Sampled motion frames with corresponding textual descriptions from the HumanML3D dataset.

Chapter 4: Contrastive Learning Approach

4.1 Methods

We adopt the contrastive learning algorithm BRIO Liu et al. (2022) which is originally applied to text summarization to the text-to-motion generation task. For the generative model, we choose TM2T Guo et al. (2022b) as the backbone.

TM2T proposes the motion token method to discretize 3D pose sequences by using a series of 1D convolutions along the temporal dimension to encode continuous sequences into motion tokens and then map them into the closest entries of a pre-defined motion token codebook. The token encoder CNN is pre-trained as an auto-encoder with a corresponding token decoder.

For conditional motion generation, TM2T utilizes the attention-based bi-directional GRU as text encoder and motion decoder and autoregressively predicts the probability distribution for the next token over all the entries in the token codebooks as $p_\phi(\hat{p}_t|\hat{P}_{1:t-1}, X)$ where the motion decoder is parameterized as ϕ. We can regard this decoding process as standard word-by-word text generation over the vocabulary. After that, the motion token sequence is transformed back to the real motion sequence using the pre-trained token decoder.

The discrete decoding process of motion tokens allows us to apply the contrastive loss introduced in the BRIO paper. For each input textual description, we generate multiple motions as candidates, the contrastive loss is computed as:

$$\mathcal{L}_{ctr} = \sum_i \sum_{j>i} \max(0, f(\hat{P}^i, P) - f(\hat{P}^j, P) + \lambda_{ij}) \tag{4.1}$$

where $\hat{P}^i, \hat{P}^j$ are two candidate motions such that $\forall i, j, i < j, S(\hat{P}^i, P) > S(\hat{P}^j, P)$, in other words, $\hat{P}^i, \hat{P}^j$ are the $i^{\text{th}}, j^{\text{th}}$ candidates ranked by the score S. P is the corresponding ground truth motion and S is our target evaluation metric to optimize.

$\lambda_{ij} = (j - i) * \lambda$ where λ is the margin hyperparameter. $f(\hat{P}^i)$ is the length-normalized estimated log-probability of the candidate:

$$f(P) = \frac{\sum_{t=1}^{T} \text{log} p_\phi(p_t | P_{1:t-1}, X)}{T^\alpha} \tag{4.2}$$

where α is the length penalty hyperparameter. The contrastive loss function gives the backbone TM2T model the ability to evaluate different motions as a discriminator. The reason is that the predicted motion probabilities are aligned with the evaluation metric S. In other words, the model assigns higher probabilities to the motions that have higher S scores.

In addition, following the analysis in the BRIO paper, we keep the original MLE loss to preserve the ability of the model as a motion generator. The final loss is computed as:

$$\mathcal{L}_{MLE} = -\sum_{t=1}^{T} \text{log} p_\phi(\hat{p}_t | \hat{P}_{1:t-1}, X) \tag{4.3}$$

$$\mathcal{L} = \mu_{mle}\mathcal{L}_{MLE} + \mu_{ctr}\mathcal{L}_{ctr} \tag{4.4}$$

where μ_{mle}, μ_{ctr} are the weights for the MLE loss and contrastive loss respectively.

In our experiment, we follow the work Lin et al. (2018) to use Dynamic Time Warped Mean Absolute Error (DTW-MAE) as our scoring function. DTW-MAE is an align-based metric that uses the dynamic time warping algorithm Salvador and Chan (2007) to align different lengths of motions and compute the mean absolute error (L1 distance) between the candidates and the references. Consider two motion sequences $P_1 = \{p_1^1, ..., p_{T_1}^1\}, P_2 = \{p_1^2, ..., p_{T_2}^2\}$, DTW-MAE returns the optimal sequence of length K of the mapping between p_i^1, p_j^2 indicated by tuples (i, j) which minimizes the total L1 distance between the mapping. Then the optimal sequence to kth mapping (i_k, j_k) is defined as:

$$D_{min}(i_k, j_k) = \min_{i_{k-1}, j_{k-1}} D_{min}(i_{k-1}, j_{k-1}) + d(i_k, j_k | i_{k-1}, j_{k-1}) \tag{4.5}$$

where d is the mean absolute error (L1 distance) between p_{i_k} and p_{j_k}. The DTW-MAE over the two entire motion sequences is computed as:

$$D(P_1, P_2) = \sum_{1 \leq k \leq K} d(i_k, j_k) \tag{4.6}$$

where $i_1 = 1, i_K = T_1$ and $j_1 = 1, j_K = T_2$.

By further fine-tuning TM2T model based on the pre-trained checkpoints using the contrastive loss, our goal is to align the probabilities of candidates predicted by the model with the target evaluation metric that we want to optimize so that the model can generate motions of higher evaluation scores with higher probabilities.

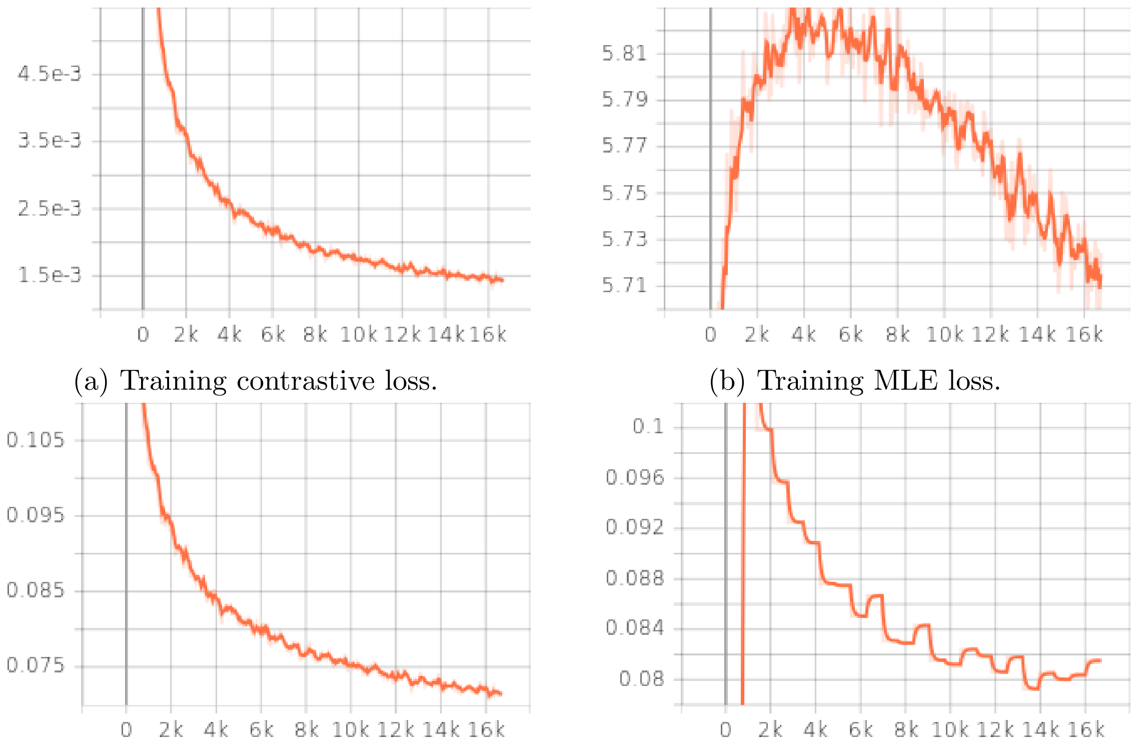
4.2 Experiments

We further fine-tune the pre-trained TM2T model using the loss in equation 4.4 with the same hyperparameter setting as BRIO $\mu_{mle} = 0.01, \mu_{ctr} = 10$ and the number of candidates $N = 10$ for each text prompt in both the train-validation set and test set of the HumanML3D dataset Guo et al. (2022a). For the backbone TM2T model, we also inherit the same architecture from the pre-trained checkpoints and apply early stopping when the validation loss starts to increase to prevent overfitting. We record all the loss terms and show the training and validation loss curves in Figure 4.1. The model structure and hyperparameter setting of the original TM2T model are shown in the appendix.

We use the DTW-MAE metric again to test the fine-tuned model on the test split and compare the results with the pre-trained model. For each textual description, we calculate the mean, max and min error values over the 10 candidate motions as the corresponding score. The average scores over the test set are shown in Table 4.1

4.3 Analysis

As shown in Figure 4.1, the contrastive loss decreases through the entire training process which indicates the gradual improvement of the alignment between the



(a) Training contrastive loss.

(b) Training MLE loss.

(c) Training loss.

(d) Validation loss.

Figure 4.1: Training and validation loss for fine-tuning TM2T model.

	Pre-trained model	Fine-tuned model
Min	**10.837**	12.291
Mean	17.451	**15.400**
Max	25.775	**19.133**

Table 4.1: Average DTW-MAE of the test set. The best results of each row are in bold font.

model predictions and evaluation scores. On the other hand, the training MLE loss first increases and then decreases with the contrastive loss. This is expected for our further fine-tuning setting because the additional contrastive loss first disrupts the pre-trained parameters through gradient descent and dominates the overall training loss. As the contrastive loss converges, its influence on the model parameters becomes less so that the MLE loss can also be optimized. For the validation loss, we see a continuously decreasing trend which indicates the ability of the fine-tuned model to align unseen motions with the DTW-MAE scores.

From Table 4.1, we see that when we calculate the mean and max DTW-MAE values for each textual description, the average result on the test set improves from $17.451, 25.775$ to $15.400, 19.133$ respectively after fine-tuning with the additional contrastive loss. The reduced average mean error indicates the overall effectiveness of our contrastive learning method and the better quality of generated motions. On the other hand, if we take the min value for each text prompt, the average test error increases from 10.837 to 12.291. Together with the decreased max error, we regard such performance differences as a good sign for reduced variance among motion candidates because the distribution of error lies closer to the mean. In other words, our fine-tuned model is able to generate motions that have higher quality in terms of DTW-MAE (closer to ground truth) with higher probabilities compared with the pre-trained model.

4.4 Discussion

Although our preliminary experiments show promising results, there is a significant limitation for our proposed contrastive learning method. This method only applies to Seq2Seq models that predict the exact probabilities of generated candidates and relies on the discrete nature of motion tokens which is often unachievable for continuous motion sequences. We notice the rapid trend recently of applying diffusion models to the text-to-motion task such as MDM Tevet et al. (2022b) and MotionDiffuse Zhang et al. (2022). In addition, as the recent work Voas et al. (2022) shows, the variational autoencoder (VAE) model T2M Guo et al. (2022a) and diffusion model MotionDiffuse Zhang et al. (2022) achieve competitive performance in human evaluation in terms of naturalness (whether the motion looks real) and faithfulness (whether the motion corresponds to the text prompt). Because either VAE which predicts the noise vector sampled from prior and posterior distribution or diffusion model which predicts the Gaussian noise does not calculate the exact log-probability for a candidate motion, there is no direct way to adopt the contrastive loss to these state-of-the-art models. We will keep working on this topic in the future to find the contrastive learning method compatible with the text-to-motion task.

Chapter 5: Differentiable Metric Approach

5.1 Methods

Most currently used evaluation metrics for text-to-motion task, for example, FID and R-precision Guo et al. (2022a) are not differentiable with respect to the network parameters. This prevents us from incorporating these metrics directly into the training objectives because the normal training process optimizes the objectives through gradient descent and computing the gradient of these complex metric functions is intractable. One solution to address the optimization problem is to develop differentiable evaluation metrics that are either novel or derived from existing ones.

Previous works Wang et al. (2019) in text-to-text domain proposed a differentiable version of the widely used metric BLEU and showed the effectiveness in machine translation and image captioning tasks. However, it did not present solid evidence that optimizing the differentiable version of this metric directly improves the model's performance on the original BLEU. In addition, as shown in our recent work Voas et al. (2022) in which we examine different evaluation metrics for the text-to-motion task, traditional coordinate error (CE) metrics like average positional error (APE), average variance error (AVE) and their variances like velocity APE (VEL APE) are still competitive with more complex learned metrics like FID and R-precision in terms of the correlation with human judgment. The advantage of these metrics is that they are originally differentiable so that they can be directly optimized through gradient descent as training objectives.

We propose a novel training loss called metric loss which is computed as either one single coordinate error (CE) or the weighted summation of multiple CE metrics. For different variants of CE metrics, our implementation of AVE and APE is based on the calculation in Ghosh et al. (2021); Voas et al. (2022). For the jth joint $P^j = \{p_1^j, ..., p_T^j\}$, the APE and AVE are computed as:

$$\text{APE}_j = \frac{1}{NT} \sum_{n \in N} \sum_{t \in T} ||p_t^j - \hat{p}_t^j||_2 \tag{5.1}$$

$$\bar{P}^j = \frac{1}{T} \sum_{t \in T} p_t^j \tag{5.2}$$

$$\sigma_j = \frac{1}{T-1} \sum_{t \in T} (p_t^j - \bar{p}^j)^2 \tag{5.3}$$

$$\text{AVE}_j = \frac{1}{N} \sum_{n \in N} ||\sigma_j - \hat{\sigma}_j||_2 \tag{5.4}$$

where N is the total number of motions in the dataset and $\bar{P}^j$ is the mean pose over T time steps. For generated motions and ground truths with different lengths, we choose T to be the minimum of the two values. σ_j is the variance of ground truth motion and $\hat{\sigma}_j$ is the variance of the generated motion. Note that APE and AVE are all differentiable with respect to the model parameters, and thus our proposed metric loss is also differentiable. For velocity and acceleration, we replace the 3D position coordinates with 3D velocity vectors by subtracting the pose at time $t + 1$ and time t and similarly for acceleration. Because the time gap is fixed, we omit the Δt in the denominator.

$$vel_t = p_{t+1} - p_t \tag{5.5}$$

$$acce_t = vel_{t+1} - vel_t \tag{5.6}$$

For the text-to-motion task, we choose our backbone model to be the variational autoencoder-based model called T2M Guo et al. (2022a). T2M achieves the best performance on motion faithfulness and the second-best performance on naturalness in human evaluation according to Voas et al. (2022).

Similar to the motion token technique in TM2T Guo et al. (2022b), T2M also uses a pre-trained motion autoencoder to transform the motion sequence to a motion snippet code sequence but replaces the Seq2Seq generator with a variational autoencoder to achieve stochastic motion generation. The VAE architecture is composed of three networks to model the generator F_θ, posterior F_ϕ and prior F_ψ. The VAE takes the context embedding c of the textual description from the text encoder as the input and generates the code snippet sequence $\hat{c}^s_{1:T}$. At each time step, the posterior and prior network sample the noise vector z_t from the approximated distribution as:

$$\mathrm{F}_\phi = \mathrm{q}_\phi(\mathrm{z_t}|\hat{\mathrm{c}}^{\mathrm{s}}_{1:\mathrm{t}}, \mathrm{c}) \tag{5.7}$$

$$\mathrm{F}_\psi = \mathrm{p}_\psi(\mathrm{z_t}|\hat{\mathrm{c}}^{\mathrm{s}}_{1:\mathrm{t}-1}, \mathrm{c}) \tag{5.8}$$

The generator takes as input the noise vector z_t, the attention vector w^{att}_t by attending the word features with the current generator hidden state and generated snippet code sequence so far $\hat{c}^s_{1:t-1}$, and output $\hat{c}^s_t$. The noise vector is sampled from the posterior distribution q_ϕ during training time while from the prior distribution p_ψ during the testing time due to the unavailable generated pose at time t. Finally, the reconstructed motion snippet sequence $\hat{c}^s_{1:T}$ is decoded back to the actual motion sequence $\hat{P}$ using the pre-trained motion autoencoder. To train the posterior F_ϕ and prior networks F_ψ in the VAE, the loss is defined as the KL-divergence between the two distributions:

$$\mathcal{L}_{KL} = \sum_{t\in T} \mathrm{KL}(\mathrm{q}_\phi(\mathrm{z_t}|\hat{\mathrm{c}}^{\mathrm{s}}_{1:\mathrm{t}}, \mathrm{c})||\mathrm{p}_\psi(\mathrm{z_t}|\hat{\mathrm{c}}^{\mathrm{s}}_{1:\mathrm{t}-1}, \mathrm{c})) \tag{5.9}$$

To train the generator network F_θ, the reconstruction losses of both motion snippet codes and motion sequence are computed as:

$$\mathcal{L}^{code}_{rec} = \sum_{t\in T} ||\hat{c}^s_t - c^s_t||_1 \tag{5.10}$$

$$\mathcal{L}^{mot}_{rec} = \sum_{t\in T} ||\hat{p}_t - p_t||_1 \tag{5.11}$$

where $\hat{c}_t^s, \hat{p}_t$ are the model generations and c_t^s, p_t are the ground truths.

Let $\mathcal{L}_{met}$ be our proposed metric loss, the final loss is calculated as:

$$\mathcal{L} = \lambda_{code}\mathcal{L}_{rec}^{code} + \lambda_{mot}\mathcal{L}_{rec}^{mot} + \lambda_{KL}\mathcal{L}_{KL} + \lambda_{met}\mathcal{L}_{met} \tag{5.12}$$

where $\lambda_{code}, \lambda_{mot}, \lambda_{met}$ are the loss weights controlling the relative magnitudes of different loss terms. The complete model structure of T2M is beyond the scope of our project and we present the architecture in the appendix.

5.2 Experiments Results and Analysis

We conduct comprehensive experiments from different perspectives to study how our proposed metric loss influences the performance of the T2M model. For all the hyperparameters except the ones related to the proposed metric loss, we follow the same setting as in T2M which is also shown in the appendix.

5.2.1 Preliminary Experiment

Our preliminary experiments investigate three different training processes:

- Fine-tune: Based on the pre-trained T2M checkpoints, we further fine-tune the model using the loss 5.12.
- Train from scratch: We train a new model with respect to the loss 5.12 from scratch.
- Fine-tune with only metric loss: We further fine-tune the pre-trained T2M checkpoint using only the metric loss $\mathcal{L}_{met}$.

One thing to note is that for the Train from scratch option, we follow the same curriculum as the original paper. The model is first optimized to generate motion snippet sequences of length 1, if the training loss converges, the predicted length adds 1 and the training process continues until the maximum length 49 is reached.

Our preliminary experiments involve 4 variants of metric loss:

- Root POS AVE
- Joint POS AVE
- Joint POS APE
- Pose VEL APE

where "Root" stands for the root position of the skeleton and "Joint" stands for all the other body joints. "POS" and "VEL" stand for the 3D position and velocity respectively. We select these variants of coordinate error because they are among the best metrics that correlate with human judgment according to Voas et al. (2022). Finally, we set the loss weights λ_{met} for Root POS AVE + Joint POS AVE, Root POS APE, Joint POS APE to 10, and Pose VEL APE to 100 during training. By selecting these loss weights, we aim to make the magnitude of different CE metrics relatively close to the original loss so that the impact of the additional metric loss is neither too large that buries the original loss nor too small that contributes little to the optimization. The corresponding evaluation results of different AVE and APE on the test set before and after training the model are shown in Table 5.1.

	Pre-train	Fine-tune	Train from scratch	Fine-tune with only metric loss
Root POS AVE + Joint POS AVE	**0.6266**	0.8039	0.7629	
Root POS APE	**0.6001**	0.7918		
Joint POS APE	**0.6501**	0.6909	0.7072	1.0643
Pose VEL APE	0.0283	**0.0265**	0.0277	

Table 5.1: Average errors of the test set using 3 training methods. The best result of each variant of metric loss (row) is in bold font.

Note that we black out the Fine-tune with only the metric loss column because we see the huge performance drop from 0.6501 to 1.0643, so we omit the results on the other models and drop this option for the rest of our experiments. For all the experiments, we find that the model achieves better performance when we fine-tune it than the other two options in general. Therefore, we choose to focus on the Fine-tune method when we do training. Another finding is that the only metric that is actually improved is the Pose VEL APE through fine-tuning or training from scratch. We also evaluate the performance on the other metrics when training to optimize Pose VEL APE and present the results in Table 5.2.

	Pre-train	Fine-tune with VEL APE	Train from scratch with VEL APE
Root POS AVE + Joint POS AVE	**0.6266**	0.7123	0.6769
Root POS APE	**0.6001**	0.6401	0.6579
Joint POS APE	**0.6501**	0.6888	0.7104
Pose VEL APE	0.0283	**0.0265**	0.2768

Table 5.2: Evaluation results of the trained models with VEL APE on the other metrics

As shown in Table 5.1, among the three training options: fine-tuning, training from scratch, and fine-tuning with only metric loss, the last option leads to the largest performance drop which indicates that the reconstruction loss and KL-divergence are essential to the VAE model to control the basic motion shapes and generate reasonable motions. On the other hand, we did not see very promising results by either fine-tuning or training from scratch using our selected CE metrics except for Pose VEL APE. When we fine-tune the model using Pose VEL APE as the metric loss, the corresponding error reduces from 0.0283 to 0.0265. However, this fine-tuned model does not achieve better performance on the other selected metrics over the pre-trained model as shown in Table 5.2, which indicates that the latent interaction between different CE metrics is complex and may not necessarily be a positive correlation.

One good sign we can see from Table 5.2 is that for the Root POS AVE + Joint POS AVE and Joint APE, the fine-tuned model with VEL APE even outperforms the models fine-tuned with the original metrics. Optimizing one metric may at the same time benefit the model performance on other metrics. This to some extent reflects the importance of the choice of metric to optimize because different CE metrics influence the overall model performance at different levels.

5.2.2 Optimizing Single Metric and Linear Combination

Our next step which is also our main focus is to systematically examine and analyze the relationship between metric losses of different choices of one single CE metric and the linear combination of multiple metric variants. In order to evaluate the performance in a more systematic way, we select the three most important CE metrics: Pose POS APE, Pose VEL APE and Pose ACCE APE from the total of 18 variants and measure the metric changes after fine-tuning. We use two simple linear combinations:

- POS APE + 10×VEL APE
- POS APE + 10×VEL APE + 10×ACCE APE

Note that VEL APE and ACCE APE are scaled up by a factor of 10 because the magnitudes of these values are small compared with POS APE. For simplicity of expression, we denote these two weighted metrics by PV APE and PVA APE respectively. In addition, we explore the loss weights that largely control the relative magnitudes of metric losses with respect to the original reconstruction losses and KL divergence. We also tune the hyperparameter, the metric loss weight λ_{met}. We follow a relatively empirical process for selecting the λ_{met} values by starting with a weight value that makes metric loss closest to the original losses and then dividing it by 10 as a smaller weight because we notice that smaller loss weight, in general, improves the overall performance. More details will be discussed in later paragraphs. For hyperparameter tuning, grid search is actually a more systematic way. However, because the original magnitudes of metric loss variants also vary, we may not directly compare the performance of different models under the same λ_{met} setting.

Finally, besides the trivial metric combination, we try to train a linear combination of coordinate error (CE) metrics using the human evaluation data in Voas et al. (2022). We have two options. We select the top-5 CE metrics with the highest Pearson's correlation score with the human evaluation of faithfulness at both by-sample level and by-model level. We train a linear regression model for each option separately and optimize the L2 distance between the weighted summation of the 5 metrics and the human scores using 5-fold validation. The weights that correspond to the highest correlation on the held-out validation data are chosen as the metric loss. The losses are denoted by LIN COMB S and LIN COMB M indicating the option that we select CE metrics. We then fine-tune the model using our linear combination metric losses and set the loss weight $\lambda_{met} = 0.01$. Table 5.3 and Table 5.4 show the selected 5 CE metrics and the trained linear combination with the corresponding correlation scores. The values of the two weighted metrics before and after fine-tuning T2M are shown in Table 5.5.

	Pearson's correlation
Joint POS APE	0.1677
Pose POS APE	0.1677
Joint POS AVE	0.1067
Joint VEL AVE	0.0635
Joint ACCE AVE	0.0504
LIN COMB S	0.2142

Table 5.3: Top-5 metrics that are best correlated with human evaluation of motion faithfulness at by-sample level and the trained linear combination.

All the evaluation results of the single metric and linear combination models mentioned above are shown in Table 5.6.

We also present some supplementary experiments in Table 5.7 that are not categorized into any slots in Table 5.6 including training from scratch with PVA APE, fine-tuning with the concatenation of POS APE, VEL APE and ACCE APE. In addition, we also tested the performance of TM2T model Guo et al. (2022b). These

	Pearson's correlation
Joint VEL APE	0.5665
Pose VEL APE	0.5665
Joint VEL AVE	0.4642
Joint ACEL APE	0.4016
Pose ACCE APE	0.4016
LIN COMB M	0.8467

Table 5.4: Top-5 metrics that are best correlated with human evaluation of motion faithfulness at by-model level and the trained linear combination.

	Pre-train	Fine-tune
LIN COMB S	2.4853	**2.4416**
LIN COMB M	1.0935	**1.0525**

Table 5.5: Values of LIN COMB S and LIN COMB M before and after fine-tuning with respect to the corresponding metric losses.

Model\Pose CE	POS APE	VEL APE	ACCE APE
Pre-train	0.6478	0.0283	0.0112
VEL APE (λ_{met}=100)	0.6865	**0.0265**	**0.0089**
VEL APE (λ_{met}=10)	0.6732	0.0277	0.0099
VEL APE (λ_{met}=1)	**0.6701**	0.0282	0.0110
POS APE (λ_{met}=1)	0.6675	**0.0275**	**0.0110**
POS APE (λ_{met}=0.1)	**0.6611**	0.0287	0.0114
PV APE (λ_{met}=1)	0.6810	0.0278	**0.0101**
PV APE (λ_{met}=0.1)	0.6666	0.0283	0.0110
PV APE (λ_{met}=0.01)	**0.6489**	**0.0276**	0.0108
PVA APE (λ_{met}=10)	0.7394	0.0283	0.0103
PVA APE (λ_{met}=1)	**0.6860**	**0.0279**	**0.0103**
LIN COMB S	**0.6362**	**0.0274**	**0.0110**
LIN COMB M	0.6373	0.0280	0.0113

Table 5.6: Evaluation results of single and linear combined metric losses and hyperparameter (λ_{met}) tuning. The best results among all variants of metric losses are marked in red. The best results of a metric loss with different loss weights are in bold font. All models are fine-tuned using the corresponding metric losses,

results are not essential to our main focus but we can draw some findings from them. More details will be discussed in the following Analysis section.

	POS APE	VEL APE	ACCE APE
PVA APE (λ_{met}=10) Train from scratch	0.73504	0.0288	0.0092
Concatenation (λ_{met}=1)	0.8574	0.0320	0.0126
TM2T	0.6967	0.0289	0.0085

Table 5.7: Evaluation results of supplementary experiments.

We draw several important findings from Table 5.6. For different choices of single CE metric POS APE vs. VEL APE, although we cannot directly improve the performance on POS APE compared with the pre-trained model, optimizing metric loss of single VEL APE can lead to the best performance on VEL APE itself and also ACCE APE. The fine-tuned model with POS APE still outperforms the one with VEL APE in terms of the corresponding metric, which indicates that optimizing the metric loss are still correlated with the performance of the corresponding CE metric. We also note that POS APE is the most difficult metric to optimize, and only PV APE ($\lambda_{met} = 0.01$) and the LIN COMB model achieves competitive or better results on it. However, we see that most fine-tuned models can reduce VEL APE and ACCE APE.

For the two trivial linear combination metrics, PV APE consistently outperforms the PVA APE when $\lambda_{met} = 1$ while the learned LIN COMB model achieves the best POS APE which outlines the importance of carefully computed weights for different metrics in the linear combination when the components of metric loss become complex which makes it harder to optimize. For the trained linear combination, we first show in Table 5.5 that both linear combination losses are actually improved through fine-tuning. The sample level model (LIN COMB S) outperforms the model level model (LIN COMB M) on all three APE metrics. We attribute the differences in model performance to the training process of the linear combination. Because we reuse the human scores from Voas et al. (2022) which only contains four models, Pearson's correlation is highly unstable when we conduct 5-fold training of the linear combination.

We notice that the loss weight λ_{met} has a large impact on the model performance. Results in Table 5.6 show that, in general, a smaller λ_{met} will lead to a better result of POS APE but slightly worse results of VEL APE and ACCE APE except for the case for PVA APE model where all the three measurements have improvement for a smaller λ_{met}. We contribute such improvement of POS APE to the smaller magnitude of metric loss compared with the original reconstruction loss and the KL divergence. We hypothesize the additional metric loss can cause disturbance to the pre-trained model parameters that are trained with respect to only the original losses. A smaller loss weight has less disturbance to the pre-trained motion dynamics with respect to human capturing data which directly corresponds to POS APE. As a result, the fine-tuned models with metric loss all to different extents diverge from the pre-trained model and have worse results in terms of POS APE except for the two LIN COMB models. The possible reason for the good performance of LIN COMB S and LIN COMB M on POS APE is that there are average variance error (AVE) metrics as part of the loss which implicitly mitigates the direct disturbance to POS APE. On the other hand, for VEL APE and ACCE APE, because they are the first and second derivatives of POS APE and are not directly associated with the original losses, we can optimize them by introducing the corresponding CE metrics to the loss.

According to Table 5.7 using the concatenation of the three components as the metric loss does not contribute to any improvement of errors because it blurs the derivatives in different orders and different influences of CE metrics towards the model. Also, the bad performance of TM2T is consistent with the human evaluation results in Voas et al. (2022).

5.2.3 Results of Non-differentiable Metrics

In order to make our model comparable with existing models, we also evaluate the fine-tuned models with the metric loss on some non-differentiable metrics including FID, Multi-Modal Distance (MM Dist) and R-Precision that are used in previous works Guo et al. (2022a,b). For R-Precision, we use the top-3 accuracy. Finally, we present the performance on our novel nearest neighbor captioning metric (NNC) Voas et al. (2022), we use the Euclidean distance (L2 Dist) and cosine distance (COS Dist) to evaluate the similarity between the input and retrieved texts. The results are shown in Table 5.8

	MM Dist ↓	FID ↓	R-Precision (Top-3) ↑	NNC (L2 Dist) ↓	NNC (COS Dist) ↓
Pre-train	3.3615	1.2552	0.7378	8.7349	0.2515
VEL APE (λ_{met}=100)	4.3744	2.2491	0.5931	8.7930	0.2631
VEL APE (λ_{met}=10)	3.5161	1.1128	0.7128	**8.7748**	0.2527
VEL APE (λ_{met}=1)	**3.3644**	**1.0108**	**0.7353**	8.7806	**0.2525**
PV APE (λ_{met}=1)	3.8834	1.1703	0.6656	8.7971	0.2566
PV APE (λ_{met}=0.1)	3.4663	1.1457	0.7221	**8.7131**	0.2512
PV APE (λ_{met}=0.01)	**3.4415**	**1.0777**	**0.7306**	8.7374	**0.2503**
PVA APE (λ_{met}=10)	5.0322	4.3125	0.4914	8.8601	0.2675
PVA APE (λ_{met}=1)	**3.8879**	**1.2392**	**0.6555**	**8.7762**	**0.2565**
LIN COMB S	**3.4347**	**0.9781**	**0.7256**	8.7393	0.2515
LIN COMB M	3.4942	1.2215	0.7163	**8.7126**	**0.2507**

Table 5.8: Evaluation results on non-differentiable metrics.

From Table 5.8, we see a promising result that many models often achieve better performance on FID and NNC. As FID and NNC are metrics that evaluate motion naturalness and faithfulness respectively, we regard these improvements as a good sign that we are able to optimize the model using the proposed metric loss on both dimensions. For different metric loss weights, we still see the general trend that a smaller λ_{met} leads to better performance on these non-differentiable metrics which is consistent with our analysis of CE metrics in Table 5.6. On the other hand, this indicates that these non-differentiable metrics are potentially more correlated with POS APE than VEL APE and ACCE APE.

5.3 Discussion

We conduct experiments from different angles to explore the effectiveness of our proposed metric loss including different CE metric variants, hyperparameter tuning of loss weight λ_{met} and draw several findings from the experiment results. However, we also want to point out the limitations.

We only cover a subset of CE metrics and their linear combination. There are still many variants whose influence on the model performance is still unknown, especially the average variance error (AVE). We exclude the AVE metrics because in general, they have a weaker correlation with human judgment according to Voas et al. (2022). However, optimizing different types of metrics can implicitly influence each other. As the good performance of the LIN COMB model shows, although AVE metrics may not be good metrics on their own, they can benefit the final results of both CE metrics and non-differentiable metrics when we include them in the linear combination. In fact, because the internal relationship between different types of CE metrics is so complicated that the best way to investigate the metric loss is to explore every variant and see the performance of single and linear combined metrics. Limited by the scope of our experiments, many of our findings may not universally apply to every different situation when factors such as CE metric choices and hyperparameter (λ_{met}) settings vary.

our experiments on metric loss with the linear combination of CE are also not sufficient. We only evaluate one type of linear combination metric loss and we train the linear combination in an empirical way without enough theoretical justification. Firstly, the CE metrics in the loss are selected by the faithfulness score while naturalness as another important aspect of motion evaluation is not considered. In addition, it is also necessary to train the linear combination with respect to faithfulness and naturalness scores at the same time such as their weighted summation. Given the already good performance of the LIN COMB model, we plan to conduct more experiments to cover a larger range of hyperparameters to search for a proper subset of CE variants to compose the linear combination metric loss.

Finally, We do not involve an important concept ”root scaling factor” introduced in Voas et al. (2022) which separately considers the root joint and body joints in the CE metric. The reason is that it may lead to more complex metric losses and make it difficult to control the influence factors of the experiments. For all the limitations discussed above, we leave them to future works.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

In our project, we review existing works in mostly the text-to-text generation domain that addresses the discrepancy between the training objectives and testing metrics. We develop algorithms including the contrastive learning approach and the differentiable metric approach to directly optimize the evaluation metrics through the training process.

In the contrastive learning approach, we show that by fine-tuning TM2T Guo et al. (2022b) model with a ranking-based contrastive loss Liu et al. (2022), we can align the model predictions of candidate motions with the target evaluation metric values. For the differentiable metric approach, we propose a novel metric loss based on the differentiable coordinate error (CE) metrics as an additional training objective. We examine variants of metric loss including different CE metrics and their linear combination and present improvement of TM2T model performance on both CE and non-differentiable metrics by training with our proposed metric loss.

6.2 Future Work

6.2.1 Human Evaluation

As the origin and ultimate goal, improving the human evaluation of motion generation is always essential to our project. Because we have achieved relatively promising results of the differentiable approach and hypothesize that directly training to optimize these automated evaluation metrics will also improve the human evaluation of the generated motions. To justify our hypothesis, we will select one model with a single CE as the metric loss and one model with the linear combination metric loss for human evaluation. We will use the fine-tuned Pose VEL APE which has the best results on Pose VEL APE and Pose ACCE APE and the fine-tuned

LIN COMB model which outperforms others on Pose POS APE to generate motions with the same textual description used in Voas et al. (2022). We will also follow the same human annotation procedure to collect the motion faithfulness and naturalness scores. These human evaluation scores can be used to compare our trained model with existing ones that are tested in our previous work Voas et al. (2022).

6.2.2 Reinforcement Learning Approach

Considering directly optimizing a specific evaluation metric through the training process, reinforcement learning is one of the most promising future directions. We can easily model the complex metrics or even human evaluations Ziegler et al. (2019) that are often non-differentiable and thus intractable to optimize through gradient descent as reward functions in the RL domain.

We propose the general setting of the reinforcement learning framework for the text-to-motion generation task as follows.

- State: At each time frame t, the state $s_t = \{c_t, \hat{P}_{1:t-1}\}$ consists of the context vector c_t and the motion generated so far $\hat{P}_{1:t-1} = \hat{p}_1, ..., \hat{p}_{t-1}$. The context vector may include the encoded input textual description and other contextual information such as the attention vector and the hidden vector from the previous time step. The specific definition of context vector depends on the text-to-motion generation model, for example, in the VAE model Guo et al. (2022a), the context vector is the concatenation of the attention vector with respect to the textual description embedding, previous hidden state and a time-to-arrival positional encoding indicating the current time step.

- Action: The action a_t to take at time t is the 3D pose representation to generate at the current time step $\hat{p}_t \in \mathbb{R}^{J \times 3}$

- Transition function: The transition function takes the previous state and the action at time t, and outputs the next state. $s_{t+1} = \text{trans}(s_t, a_t)$ where $s_{t+1} =$

$\{c_{t+1}, \hat{P}_{1:t}\}$. For the generated motion sequence so far, we just concatenate $\hat{p}_t$ with $\hat{P}_{1:t-1}$. The transition for the context vector also depends on the generative model type. In the VAE model, the hidden state and TTA encoding move from time t to $t-1$ while the text embedding stays unchanged.

- Policy: We want to adopt existing text-to-motion models as the policy network which takes the current state as input and predicts deterministically the next pose (e.g. Seq2Seq models) or stochastically the distribution to sample the next pose (e.g. VAE models).
- Reward: The rewards are the scores given by the evaluation metrics that we want to optimize like FID Heusel et al. (2017). Under this setting, rewards can only be received after the entire motion sequence has been generated.

Note that for text-to-motion generation, both state space and action space are continuous, we want to use the Deep Deterministic Policy Gradient (DDPG) algorithm Lillicrap et al. (2016) which is based on the actor-critic framework in which both actor (Policy) and critic (Value) are represented as deep neural networks.

Due to the lack of related work on RL for language-conditioned multi-modal generative models and the complexity of implementation, our proposed idea on RL methods is still at the beginning stage. Future works will include fitting the text-to-motion model into the DDPG settings. More technical details especially the policy gradient computation need further investigation.

Appendix A: Text-to-Motion Generation Model

We show the detailed network architectures and hyperparameter settings of the backbone models TM2T Guo et al. (2022b) for the contrastive learning method and T2M Guo et al. (2022a) for the differentiable metric method.

A.1 TM2T

A.1.1 Hyperparameter Setting

The hyperparameter settings for the TM2T model retrieved from the original implementation are shown as follows:

batch_size: 32
codebook_size: 1024
d_inner_hid: 2048
d_k: 64
d_model: 512
d_v: 64
dim_mot_hid: 1024
dim_txt_hid: 512
dim_vq_dec_hidden: 1024
dim_vq_enc_hidden: 1024
dim_vq_latent: 1024
dropout: 0.1
early_or_late: early
label_smoothing: False
lambda_beta: 1
lambda_m2t: 1.0
lr: 0.0002

```
m2t_v3: False
max_epoch: 100
max_text_len: 20
n_dec_layers: 6
n_down: 2
n_enc_layers: 6
n_head: 8
n_mot_layers: 1
n_resblk: 3
num_candidates: 10
proj_share_weight: False
q_mode: cmt
start_m2t_ep: 0
text_aug: False
tf_ratio: 0.4
top_k: -1
unit_length: 4
```

A.1.2 Appendix section

The detailed TM2T model architecture as printed out by the original implementation is shown as follows:

```
Seq2SeqText2MotScorer(
(text_encoder): TextEncoderBiGRU(
(input_emb): Linear(in_features=300, out_features=512, bias=True)
(gru): GRU(512, 512, batch_first=True, bidirectional=True)
)
(motion_decoder_step): MotionEarlyAttDecoder(
(input_emb): Embedding(1027, 1024)
(z2init): Linear(in_features=1024, out_features=1024, bias=True)
(gru): ModuleList(
(0): GRUCell(1024, 1024)
)
(att_layer): AttLayer(
(W_q): Linear(in_features=1024, out_features=1024, bias=True)
(W_k): Linear(in_features=1024, out_features=1024, bias=False)
(W_v): Linear(in_features=1024, out_features=1024, bias=True)
(softmax): Softmax(dim=1)
)
(att_linear): Sequential(
(0): Linear(in_features=2048, out_features=1024, bias=True)
(1): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
(2): LeakyReLU(negative_slope=0.2, inplace=True)
)
(trg_word_prj): Linear(in_features=1024, out_features=1027, bias=False)
)
)
```

A.2 T2M

A.2.1 Hyperparameter Setting

The hyperparameter settings for the T2M model retrieved from the original implementation are shown as follows:

```
batch_size: 32
dim_att_vec: 512
dim_dec_hidden: 1024
dim_movement_dec_hidden: 512
```

```
dim_movement_enc_hidden: 512
dim_movement_latent: 512
dim_pos_hidden: 1024
dim_pri_hidden: 1024
dim_text_hidden: 512
dim_z: 128
early_stop_count: 3
estimator_mod: bigru
feat_bias: 5
lambda_kld: 0.01
lambda_metric: 0.0001
lambda_rec_mot: 1
lambda_rec_mov: 1
lr: 0.0002
max_sub_epoch: 50
max_text_len: 20
n_layers_dec: 1
n_layers_pos: 1
n_layers_pri: 1
text_enc_mod: bigru
tf_ratio: 0.4
unit_length: 4
```

A.2.2 Model Architecture

The detailed T2M model architecture as printed out by the original implementation is shown as follows:

```
TextEncoderBiGRU(
(pos_emb): Linear(in_features=15, out_features=300, bias=True)
```

```
(input_emb): Linear(in_features=300, out_features=512, bias=True)
(gru): GRU(512, 512, batch_first=True, bidirectional=True)
)
TextDecoder(
(emb): Sequential(
(0): Linear(in_features=1024, out_features=1024, bias=True)
(1): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
(2): LeakyReLU(negative_slope=0.2, inplace=True)
)
(gru): ModuleList(
(0): GRUCell(1024, 1024)
)
(z2init): Linear(in_features=1024, out_features=1024, bias=True)
(positional_encoder): PositionalEncoding()
(mu_net): Linear(in_features=1024, out_features=128, bias=True)
(logvar_net): Linear(in_features=1024, out_features=128, bias=True)
)
TextDecoder(
(emb): Sequential(
(0): Linear(in_features=1536, out_features=1024, bias=True)
(1): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
(2): LeakyReLU(negative_slope=0.2, inplace=True)
)
(gru): ModuleList(
(0): GRUCell(1024, 1024)
)
(z2init): Linear(in_features=1024, out_features=1024, bias=True)
(positional_encoder): PositionalEncoding()
(mu_net): Linear(in_features=1024, out_features=128, bias=True)
```

```
(logvar_net): Linear(in_features=1024, out_features=128, bias=True)
)
TextVAEDecoder(
(emb): Sequential(
(0): Linear(in_features=1152, out_features=1024, bias=True)
(1): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
(2): LeakyReLU(negative_slope=0.2, inplace=True)
)
(z2init): Linear(in_features=1024, out_features=1024, bias=True)
(gru): ModuleList(
(0): GRUCell(1024, 1024)
)
(positional_encoder): PositionalEncoding()
(output): Sequential(
(0): Linear(in_features=1024, out_features=1024, bias=True)
(1): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
(2): LeakyReLU(negative_slope=0.2, inplace=True)
(3): Linear(in_features=1024, out_features=512, bias=True)
)
)
AttLayer(
(W_q): Linear(in_features=1024, out_features=512, bias=True)
(W_k): Linear(in_features=1024, out_features=512, bias=False)
(W_v): Linear(in_features=1024, out_features=512, bias=True)
(softmax): Softmax(dim=1)
)
```

Works Cited

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

Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. On the weaknesses of reinforcement learning for neural machine translation. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=H1eCw3EKvH`.

Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. Synthesis of compositional animations from textual descriptions. In *ICCV*, pages 1376–1386. IEEE, 2021.

Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion. In *Proceedings of the 28th ACM International Conference on Multimedia*. ACM, Oct 2020.

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 5142–5151. IEEE, 2022a.

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 (35)*, volume 13695 of *Lecture Notes in Computer Science*, pages 580–597. Springer, 2022b.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL `http://arxiv.org/abs/1509.02971`.

Angela S. Lin, Lemeng Wu, Rodolfo Corona, Kevin W. H. Tai, Qixing Huang, and Raymond J. Mooney. Generating animated videos of human activities from natural language descriptions. In *Proceedings of the Visually Grounded Interaction and Language Workshop at NeurIPS*, 2018.

Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 873–881. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.100. URL `https://doi.org/10.1109/ICCV.2017.100`.

Yixin Liu, Pengfei Liu, Dragomir R. Radev, and Graham Neubig. BRIO: bringing order to abstractive summarization. *CoRR*, abs/2203.16804, 2022. doi: 10.48550/arXiv.2203.16804. URL `https://doi.org/10.48550/arXiv.2203.16804`.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*, 2002.

Mathis Petrovich, Michael J. Black, and Gül Varol. TEMOS: generating diverse human motions from textual descriptions. In *ECCV (22)*, volume 13682 of *Lecture Notes in Computer Science*, pages 480–497. Springer, 2022.

Matthias Plappert, Christian Mandery, and Tamim Asfour. Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks. *Robotics and Autonomous Systems*, 109:13–26, 2018. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2018.07.006. URL https://www.sciencedirect.com/science/article/pii/S0921889017306280.

Jinwei Qi and Yuxin Peng. Cross-modal bidirectional translation via reinforcement learning. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 2630–2636. ijcai.org, 2018. doi: 10.24963/ijcai.2018/365. URL https://doi.org/10.24963/ijcai.2018/365.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.06732.

Stan Salvador and Philip Chan. Fastdtw: Toward accurate dynamic time warping in linear time and space. In *Intelligent Data Analysis 11.5 (2007): 561-580*, 2007.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. Bleurt: Learning robust metrics for text generation, 2020. URL https://arxiv.org/abs/2004.04696.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. doi: 10.18653/v1/p16-1159. URL https://doi.org/10.18653/v1/p16-1159.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016. doi: 10.1109/CVPR.2016.308.

Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. *arXiv preprint arXiv:2203.08063*, 2022a.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human motion diffusion model, 2022b. URL `https://arxiv.org/abs/2209.14916`.

Jordan Voas, Yili Wang, Qixing Huang, and Raymond. J. Mooney. What is the best automated metric for text to motion generation? In *SIGGRAPH, under review*, 2022.

Wentao Wang, Zhiting Hu, Zichao Yang, Haoran Shi, and Eric P. Xing. Differentiable expected BLEU for text generation, 2019. URL `https://openreview.net/forum?id=S1x2aiRqFX`.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3612–3621, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1397. URL `https://aclanthology.org/D18-1397`.

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *CoRR*, abs/2208.15001, 2022.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2019. URL `https://arxiv.org/abs/1904.09675`.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019. URL `https://arxiv.org/abs/1909.08593`.

Vita

Yili Wang earned a Bachelor of Engineering degree in Computer Science from The Hong Kong University of Science and Technology and was awarded the First Class Honors and the Academic Achievement Medal. He is now a Master of Science student in the Computer Science Department at The University of Texas at Austin and will graduate in May 2023.

Address: ywang98@utexas.edu

This thesis was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.