

This appendix provides the following supplementary information:

- Section A: A detailed explanation of the SAN-based discriminator as an alternative to GAN mentioned in Section 3.2
- Section B: An in-depth description of the editing method introduced in Section 3.4
- Section C: Details of the experimental settings used in Sections 4
- Section D: Definitions of evaluation metrics for the text-to-motion task
- Section E: Additional experiments on the performance of the proposed method

## A. SAN-based discriminator

Takida et al. [35] incorporated a sliced optimal transport perspective into a general GAN and established a new framework, slicing adversarial network (SAN). Following this work, we first decompose the discriminator  $f_\phi$  into the last linear layer and the remaining neural part, denoted as  $\mathbf{w} \in \mathbb{R}^{d_w}$  and  $h_\varphi: \mathbb{R}^{N \times L} \rightarrow \mathbb{R}^{d_w}$  with  $d_w \in \mathbb{N}$ , respectively. This decomposition provides an interpretation of the discriminator: the neural function  $h_\varphi$  maps motion sequences to non-linear features, and then the linear layer  $\mathbf{w}$  projects them into scalars. As a preliminary step for applying SAN to the adversarial training in Section 3.2.1, we normalize  $\mathbf{w}$  using its norm, resulting in  $\omega = \mathbf{w}/\|\mathbf{w}\|_2$ , which indicates the direction of the projecting. Now, the discriminator is represented in the form of an inner product as  $f_\phi(\mathbf{x}) = \omega^\top h_\varphi(\mathbf{x})$ , and its parameter is  $\phi = \{\varphi, \omega\}$ .

The prior work has shown that the discriminator obtained from the optimal solution of  $\omega$  in the hinge loss (5) does not guarantee gradients that make the generated distribution close to the data distribution. To address this issue, we adopt the SAN maximization problem instead of Equation (5). Specifically, we optimize the neural part  $\varphi$  with the original hinge loss, while applying the Wasserstein GAN loss [1] to the direction  $\omega$ . The modified maximization objective is formulated as

$$\mathcal{L}_{\text{SAN}}(\phi; \psi, \eta, \mathbf{x}) = \mathcal{L}_{\text{GAN}}(\{\varphi, \omega^-\}; \psi, \eta, \mathbf{x}) + \omega^\top (h_{\varphi^-}(\mathbf{x}) - \mathbb{E}_{q_\eta(z|\mathbf{x})}[h_{\varphi^-}(g_\psi(z))]), \quad (12)$$

where  $(\cdot)^-$  indicates a stop gradient operator. The second term in Equation (12) induces the direction that best discriminates between the real and generated sample sets in the feature space. We employ the same minimization objective as defined in Equation (6).

## B. Editing procedure on guided generation

In Algorithm 1, we show the specific procedure for guided generation described in Section 3.4. Note that we apply a time-travel technique to the update rule in Section 3.4 as in [14, 25, 38, 42] to achieve better editing results. This technique adds noise after each gradient descent step to implicitly perform a multi-step optimization of minimizing the distance measuring function  $\mathcal{L}(\cdot, \mathbf{y})$  and lead to an improvement in the editing quality.

## C. Implementation Details

**Training setup for stage 1 model:** During the training, motion sequences are segmented into lengths of  $L = 64$ . We use AdamW optimizer and batch size of 128. The hyperparameters in Eq. (4) and (6) are set as  $\lambda_{\text{act}} = 1.0$ ,  $\lambda_{\text{reg}} = 1.0 \times 10^{-4}$ , and  $\lambda_{\text{adv}} = 1.0 \times 10^{-3}$ . Our model  $\{q_\eta, g_\psi, f_\psi\}$  are trained with a simple multi-step learning late, the first 10,000 iterations with a learning rate of  $2.0 \times 10^{-4}$ , and latter 5,000 with a learning rate of  $2.0 \times 10^{-5}$  in the case of HumanML3D dataset [12]. Note that in Section E, we also conducted training and evaluation on a different dataset, the KIT-ML dataset [30]. For this dataset, the model was trained for the first 10,000 iterations with a learning rate of  $5.0 \times 10^{-5}$ , followed by an additional 5,000 iterations with a reduced learning rate of  $5.0 \times 10^{-6}$ . In addition, we replace the reconstruction loss in (4) with smooth  $\ell_1$  loss (known as the special case of Huber loss) function and introduce a position enhancement term, following the technique in [43].

**Training setup for stage 2 model:** During the training, we use the AdamW optimizer with a batch size of 64. The model  $\epsilon_\theta$  is trained for 10,000 epochs with a cosine annealing learning rate schedule starting at  $1.0 \times 10^{-4}$ , including a warm-up phase. For inference, we adopt DDIM with 50 sampling steps and employ a trailing strategy. The classifier-free guidance scale  $s$  in Eq. (17) is set to  $s = 11$  for HumanML3D and  $s = 7$  for the KIT-ML dataset. Notably, we observed in Section E that the performance is significantly influenced by the scale  $s$ .

## D. Evaluation metrics details

We provide more details of evaluation metrics in Section 4.1. We use five metrics to quantitatively evaluate text-to-motion models. These metrics are calculated based on motion and text features extracted with pre-trained networks. More specifically, we utilize the motion encoder and text encoder provided in [12]. We denote ground-truth motion features, generated

**Algorithm 1** Guided Generation for Motion Editing

---

**Require:** motion control signal  $\mathbf{y}$ , output of text encoder  $\tau(c)$ , estimator in latent diffusion  $\epsilon_\theta(\cdot, t, \tau(c))$ , distance measuring function  $\mathcal{L}(\cdot; \mathbf{y})$ , pre-defined parameter  $\bar{\alpha}_t$ , time-dependent step-size  $\rho_t$ , and the repeat times of time-travel of each step  $\{r_1, \dots, r_T\}$ .

```

1:  $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:   for  $i = r_t, \dots, 1$  do
4:      $\epsilon_t \sim \mathcal{N}(0, \mathbf{I})$ 
5:      $\mathbf{z}_{0|t} = \frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{z}_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(\mathbf{z}_t, t, \tau(c)))$ 
6:      $\mathbf{z}_{0|t} = \mathbf{z}_{0|t} - \rho_t \nabla_{\mathbf{z}_{0|t}} \mathcal{L}_{\text{Motion}}(g_\psi(\mathbf{z}_{0|t}); \mathbf{y})$ 
7:      $\mathbf{z}_{t-1} = \sqrt{\bar{\alpha}_t}\mathbf{z}_{0|t} + \sqrt{1 - \bar{\alpha}_t - \sigma_t^2}\epsilon_\theta(\mathbf{z}_t, t, \tau(c)) + \sigma_t\epsilon_t$ 
8:     if  $i > 1$  then
9:        $\epsilon'_t \sim \mathcal{N}(0, \mathbf{I})$ 
10:       $\mathbf{z}_t = \sqrt{\alpha_t}\mathbf{z}_{t-1} + \sqrt{1 - \alpha_t}\epsilon'_t$ 
11:     end if
12:   end for
13: end for
14: return  $g_\psi(\mathbf{z}_0)$  ▷ edited motion generation sample

```

---

motion features, and text features as  $f_{\text{gt}} = \mathcal{E}_M(x_{\text{gt}}) \in \mathbb{R}^{512}$ ,  $f_{\text{pred}} = \mathcal{E}_M(x_{\text{pred}}) \in \mathbb{R}^{512}$ , and  $f_{\text{text}} = \mathcal{E}_T(c) \in \mathbb{R}^{512}$ , where  $\mathcal{E}_M(\cdot)$  and  $\mathcal{E}_T(\cdot)$  represent the motion encoder and the text encoder, respectively. We explain the five metrics below.

**R-Precision:** R-Precision evaluates semantic alignment between input text and generated motion in a sample-wise manner. Given one motion sequence and 32 text descriptions (one ground truth and thirty-one randomly selected mismatched descriptions), we rank the Euclidean distances between the motion and text embeddings. Then, we compute the average accuracy at top-1, top-2, and top-3 places.

**Fréchet Inception Distance (FID):** FID evaluates the overall motion quality by measuring the distributional difference between the motion features of the generated motions and those of real motions. We obtain FID by

$$\text{FID} := \|\mu_{\text{gt}} - \mu_{\text{pred}}\|_2^2 - \text{tr}(\Sigma_{\text{gt}} + \Sigma_{\text{pred}} - (2\Sigma_{\text{gt}}\Sigma_{\text{pred}})^{\frac{1}{2}}), \quad (13)$$

where  $\mu_{\text{gt}}$  and  $\mu_{\text{pred}}$  are the mean of  $f_{\text{gt}}$  and  $f_{\text{pred}}$ , respectively,  $\Sigma_{\text{gt}}$  and  $\Sigma_{\text{pred}}$  are their corresponding covariance matrices, and  $\text{tr}$  denotes the trace of a matrix.

**Multi-modal distance (MMDist):** MMDist gauges sample-wise semantic alignment between input text and generated motion as well. MMDist is computed as the average Euclidean distance between the motion feature of each generated motion and the text feature of its corresponding description in the test set. Precisely, given  $N$  randomly generated samples, it computes the average Euclidean distance between each text feature and its corresponding generated motion feature:

$$\text{MMDist} := \frac{1}{N} \sum_{i=1}^N \|f_{\text{pred},i} - f_{\text{text},i}\|_2^2, \quad (14)$$

where  $f_{\text{pred},i}$  and  $f_{\text{text},i}$  are the features of the  $i$ -th text-motion pair.

**Diversity:** Diversity measures the variance of the generated motions across the test set. We randomly sample  $2S_d$  motions and extract motion features  $\{f_{\text{pred},1}, \dots, f_{\text{pred},2S_d}\}$  from the motions. Then, those motion features are grouped into the same size of two subsets,  $\{v_1, \dots, v_{S_d}\}$  and  $\{v'_1, \dots, v'_{S_d}\}$ . The Diversity for this set of motions is defined as

$$\text{Diversity} := \frac{1}{S_d} \sum_{i=1}^{S_d} \|v_i - v'_i\|_2^2. \quad (15)$$

$S_d = 300$  is used in our experiments, following previous works.

**Multi-modality (MModality):** MModality measures how much the generated motions diversify within each text description. Given a set of motions with  $K$  text descriptions, we randomly sample  $2S_l$  motions corresponding to the  $k$ -th text description and extract motion features  $\{f_{\text{pred},k,1}, \dots, f_{\text{pred},k,2S_l}\}$ . Then, those motion features are grouped into the same size of two subsets,  $\{v_{k,1}, \dots, v_{k,S_l}\}$  and  $\{v'_{k,1}, \dots, v'_{k,S_l}\}$ . The MModality for this motion set is formalized as

$$\text{MModality} = \frac{1}{K \cdot S_l} \sum_{k=1}^K \sum_{i=1}^{S_l} \|v_{k,i} - v'_{k,i}\|_2^2. \quad (16)$$

$S_l = 10$  is used in our experiments, following previous works.

## E. Additional experiments

### E.1. Performance comparison on the KIT-ML dataset

We evaluate our method not only on the HumanML3D dataset presented in Section 4.1 but also on the KIT-ML dataset [30], comparing its performance against other text-to-motion generation methods. The KIT-ML dataset includes 3,911 human motion sequences and 6,278 text descriptions derived from the KIT [27] and CMU [5] datasets. These data are split into training, validation, and test sets with proportions of 80%, 5%, and 15%, respectively. Consistent with Section 4.1, the evaluated methods are categorized into three groups: (i) those using VQ-based latent representations (discrete), (ii) those employing data-space diffusion models (continuous, raw data), and (iii) those utilizing VAE-based latent representations (continuous, latent). As shown in Table 5, MoLA achieves the best performance in terms of R-Precision and MMDist, particularly among continuous-based methods. However, it does not outperform methods like MLD in terms of FID. Comparing this with Table 1, we observe a discrepancy in the FID trends across the two datasets, suggesting there is room for improvement in this aspect of MoLA.

Category	Method	R-Precision $\uparrow$			FID $\downarrow$	MMDist $\downarrow$	Diversity $\rightarrow$	MModality $\uparrow$
		Top-1	Top-2	Top-3				
N/A	Real motion data	0.424 $\pm$ .005	0.649 $\pm$ .006	0.779 $\pm$ .006	0.031 $\pm$ .004	2.788 $\pm$ .012	11.08 $\pm$ .097	-
Discrete	M2DM [21]	0.416 $\pm$ .004	0.628 $\pm$ .004	0.743 $\pm$ .004	0.515 $\pm$ .029	3.015 $\pm$ .017	11.417 $\pm$ .97	3.325 $\pm$ .37
	AttT2M [47]	0.413 $\pm$ .006	0.632 $\pm$ .006	0.751 $\pm$ .006	0.870 $\pm$ .039	3.039 $\pm$ .021	10.96 $\pm$ .123	2.281 $\pm$ .047
	T2M-GPT [43]	0.416 $\pm$ .006	0.627 $\pm$ .006	0.745 $\pm$ .006	0.514 $\pm$ .029	3.007 $\pm$ .029	10.921 $\pm$ .108	1.570 $\pm$ .039
	MoMask [13]	0.433 $\pm$ .007	0.656 $\pm$ .005	0.781 $\pm$ .005	0.204 $\pm$ .011	2.779 $\pm$ .022	-	1.131 $\pm$ .043
	DiverseMotion [24]	0.416 $\pm$ .005	0.637 $\pm$ .008	0.760 $\pm$ .011	0.468 $\pm$ .098	2.892 $\pm$ .041	10.873 $\pm$ .101	2.062 $\pm$ .079
	MMM [29]	0.381 $\pm$ .005	0.590 $\pm$ .006	0.718 $\pm$ .005	0.429 $\pm$ .019	3.146 $\pm$ .019	10.633 $\pm$ .097	1.105 $\pm$ .026
	ParCO [48]	0.430 $\pm$ .004	0.649 $\pm$ .007	0.772 $\pm$ .008	0.453 $\pm$ .027	2.820 $\pm$ .028	10.95 $\pm$ .094	1.245 $\pm$ .022
	BAMM [28]	0.438 $\pm$ .009	0.661 $\pm$ .009	0.788 $\pm$ .005	0.183 $\pm$ .013	2.723 $\pm$ .026	11.008 $\pm$ .094	1.609 $\pm$ .065
Continuous (raw data)	MotionDiffuse [45]	0.417 $\pm$ .004	0.621 $\pm$ .004	0.739 $\pm$ .004	1.954 $\pm$ .062	2.958 $\pm$ .005	11.10 $\pm$ .143	0.730 $\pm$ .013
	MDM [36]	0.164 $\pm$ .004	0.291 $\pm$ .004	0.396 $\pm$ .004	0.497 $\pm$ .021	9.191 $\pm$ .022	10.847 $\pm$ .109	1.907 $\pm$ .214
	Fg-T2M [37]	0.418 $\pm$ .005	0.626 $\pm$ .004	0.745 $\pm$ .004	0.571 $\pm$ .047	3.114 $\pm$ .015	10.93 $\pm$ .083	1.019 $\pm$ .029
Continuous (latent)	MLD [3]	0.390 $\pm$ .008	0.609 $\pm$ .008	0.734 $\pm$ .007	<b>0.404<math>\pm</math>.027</b>	3.204 $\pm$ .027	10.80 $\pm$ .117	<b>2.192<math>\pm</math>.071</b>
	MotionLCM [6]	-	-	-	-	-	-	-
	MoLA (ours)	<b>0.432<math>\pm</math>.008</b>	<b>0.655<math>\pm</math>.008</b>	<b>0.770<math>\pm</math>.004</b>	0.529 $\pm$ .056	<b>2.942<math>\pm</math>.053</b>	11.129 $\pm$ .158	1.789 $\pm$ .174

Table 5. Comparison with state-of-the-art methods on KIT-ML dataset. Note that discrete representations do not allow for training-free motion editing; therefore, methods based on VQ-based latent representations (Discrete) are **grayed out**. The best scores for each metric in the methods using VAE-based latent representations (Continuous (latent)) are highlighted in **bold**.

### E.2. Ablation of classifier-free diffusion guidance on stage 2

As explained in Section 3.3, we employ a classifier-free diffusion guidance technique [15]. In training, we randomly drop the condition  $\tau(c)$  with a probability of 10% and train both the conditional model  $\epsilon_\theta(z_t, t, \tau(c))$  and the unconditional model  $\epsilon_\theta(z_t, t, \emptyset)$ . In inference, the two predictions are linearly combined as follows:

$$\epsilon_\theta(z_t, t, c) = s \cdot \epsilon_\theta(z_t, t, \tau(c)) + (1 - s) \cdot \epsilon_\theta(z_t, t, \emptyset), \quad (17)$$

where  $s$  is the guidance scale, and  $s > 1$  can amplify the effect of the guidance. We investigate how the performance of our method changes with different values of the hyperparameter  $s$ . From Figure 6, we observe that the best performance in FID is achieved at  $s = 11$  on the HumanML3D dataset. Consequently, we adopt  $s = 11$  in Section 4.1.

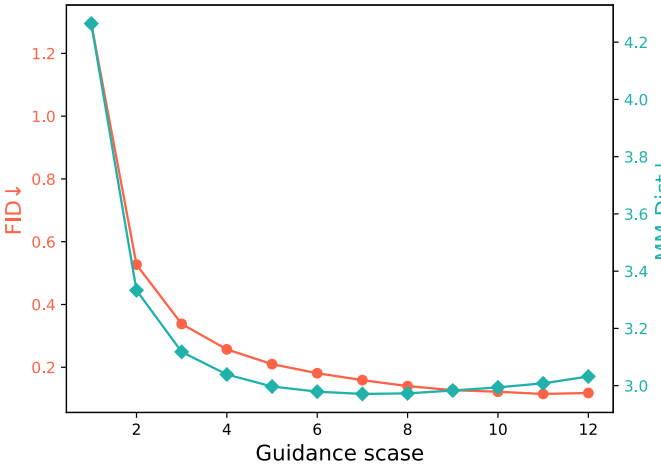


Figure 6. Evaluation sweep over guidance scale  $s$