A Appendix

Video. We provide a supplemental video, which we encourage the reviewer to watch since motion is critical in our results, and this is hard to convey in a static document.

Code and Model. The code, trained model, and re-targeted 100STYLE datasets will be made publicly available upon acceptance.

A.1 Pseudo Code

Algorithm 1 SMooDi's inference

Require: A motion diffusion model M with parameters θ_M, a style adaptor model A with parameters θ_A, style motion sequence $\boldsymbol{s}$ (if any), content texts $\boldsymbol{c}$ (if any).

$\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ # Sample from pure Gaussian distribution
for all t from T to 1 **do**
  $\{\boldsymbol{r}\} \leftarrow A(\boldsymbol{z}_t, t, \boldsymbol{c}, \boldsymbol{s}; \theta_A)$ # **Style Adaptor model**
  $\epsilon_t \leftarrow M(\boldsymbol{x}_t, t, \boldsymbol{c}, \{\boldsymbol{r}\}; \theta_M)$ # Model diffusion model
  for all k from 1 to K **do** # **Classifier-based style guidance**
    $\epsilon_t = \epsilon_t + \tau \nabla_{\boldsymbol{z}_t} G(\boldsymbol{z}_t, t, \mathbf{s})$
  end for
  $\boldsymbol{z}_{t-1} \sim \mathcal{S}\left(\boldsymbol{z}_t, \epsilon_t, t\right)$ # $S(\cdot, \cdot, \cdot)$ represents the DDIM sampling method [10].
end for
$\boldsymbol{x}_0 = \mathbf{D}(\boldsymbol{z}_0)$
return $\boldsymbol{x}_0$

A.2 Motion Style Transfer

This task involves taking a content motion sequence along with a style motion sequence and then generating a stylized motion sequence. We treat motion style transfer as one of our downstream applications and can enable SMooDi to support it without additional training. Firstly, we adopt the deterministic DDIM reverse process [43] to obtain the noised latent code $\boldsymbol{z}_T^{Inv}$ for the content motion sequence. The reverse process can be represented at step t as:

$$\boldsymbol{z}_{t+1} = \sqrt{\frac{\alpha_{t+1}}{\alpha_t}} \left(\boldsymbol{z}_t + \left(\sqrt{\frac{1}{\alpha_{t+1}} - 1} \right) - \left(\sqrt{\frac{1}{\alpha_t} - 1} \right) \right) \cdot \varepsilon_\theta(\boldsymbol{z}_t; t, \boldsymbol{c}, \emptyset), \tag{8}$$

where α represents the noise scale. $\boldsymbol{z}_T^{Inv}$ can be obtained at the last reverse step T. We substitute $\boldsymbol{z}_T$, which is initially from a pure Gaussian distribution, with the DDIM-reversed latent $\boldsymbol{z}_T^{Inv}$ in Alg. 1 and adhere to the same inference procedure to integrate the style condition into the motion content sequence throughout the denoising steps. Because there are fewer denoising steps compared to the stylized text2motion process, we slightly increase the weights of each style guidance. Specifically, the number of denoising steps is 30, $w_s = 6.5$ and $\tau = -0.4$

A.3 Implementation details

Training details. Our framework is implemented in PyTorch and trained on a single NVIDIA A5000 GPU. We use a batch size of 64, train for 50 epochs, and use the AdamW optimizer [26] with a learning rate of 1e-5. Training takes about 1 hour on a single A5000 GPU, totaling 3700 iterations. During training, we optimize the style adaptor while keeping the parameters of MLD frozen. Furthermore, to learn both the unconditioned and conditioned models simultaneously during training, we randomly set the content text $\boldsymbol{c} = \emptyset$ and mask out the style motion sequence $\boldsymbol{s}$ in the time dimension by 10%. The number of diffusion steps is $1K$ during training while 50 during interfering. The weight of classifier-free content guidance w_c is set to 7.5, classifier-free style guidance w_s is set to 1.5, and classifier-based style guidance τ is set to -0.2.

Model details. We select MLD [6] as our pre-trained motion diffusion model and use its pre-trained weights to initialize both MLD and our style adaptor. The style adaptor is composed of 4 Transformer Encoder blocks. The input process, as shown in Fig. 3, primarily involves a CLIP model [38] to encode the content text $\boldsymbol{c}$ into text embeddings, and linear layers to project the timestep t into time embeddings. These text embeddings are then added to the time embeddings and concatenated with the noisy latent $\boldsymbol{z}_t$, serving as input to the subsequent Transformer Encoder in the latent diffusion model. The style encoder, as illustrated in Fig. 3, primarily consists of a single Transformer Encoder designed to encode the style motion sequences $\boldsymbol{s}$ into style embeddings. These style embeddings are then added to the concatenated embeddings from the input process and subsequently fed into the next Transformer Encoder within the style adaptor.

Style Function details. We opt to first train a style classifier, which consists of a one-layer Transformer block, on the 100STYLE dataset for 100 epochs, using ground-truth style labels for supervision. Then, we omit the last fully connected layer to serve as our style function.

Baseline details. Due to the baselines being trained on a small style motion dataset and using different skeletons, their released pre-trained weights cannot be directly utilized. We leverage the source code from Motion Puzzle [19] and Aberman et al. [1] to implement their methods on the combined dataset, HumanML3D + 100STYLE. For a fair comparison, we replace their 4D rotation with our 6-D rotation-based feature [65]. Given the requirement for style-labeled motion data in Aberman et al. [1], we follow the same process from Motion Puzzle [19] to allow Aberman et al.'s approach to bypass this constraint. Because these baselines are trained from scratch, we increased their training iterations to five times more than ours.

Dataset details. Due to some style labels in the 100STYLE dataset inherently containing content meanings, like 'jump' and 'kick', which may conflict with the content text in the HumanML3D dataset. For example, style motion about 'kick' will conflict with content text 'a person walks forward and then backward.' To fairly compute the SRA metric, we follow [24] to categorize style labels in the 100STYLE dataset into six groups: character (CHAR), personality (PER),

emotion (EMO), action (ACT), objective (OBJ), and motivation (MOT). Notably, the 'ACT' group contains content meaning; we exclude the 'ACT' group style motion when computing the SRA metric for content text from the HumanML3D dataset. It is worth noting that we use all categories of style motion during training. Table. 4 is the detailed grouping of style labels in the 100STYLE dataset.

Table 4: The detailed grouping of style labels in the 100STYLE dataset.

Category	Label
CHAR	Aeroplane, Cat, Chicken, Dinosaur, Fairy, Monk, Morris, Penguin, Quail, Roadrunner, Robot, Rocket, Star, Superman, Zombie (15)
PER	Balance, Heavyset, Old, Rushed, Stiff (5)
EMO	Angry, Depressed, Elated, Proud (4)
ACT	kimbo, ArmsAboveHead, ArmsBehindBack, ArmsBySide, ArmsFolded, BeatChest, BentForward, BentKnees, BigSteps, BouncyLeft, BouncyRight, CrossOver, FlickLegs, Followed, GracefulArms, HandsBetweenLegs, HandsInPockets, HighKnees, KarateChop, Kick, LeanBack, LeanLeft, LeanRight, LeftHop, LegsApart, LimpLeft, LimpRight, LookUp, Lunge, March, Punch, RaisedLeftArm, RaisedRightArm, RightHop, Skip, SlideFeet, SpinAntiClock, SpinClock, StartStop, Strutting, Sweep, Teapot, Tiptoe, TogetherStep, TwoFootJump, WalkingStickLeft, WalkingStickRight, Waving, WhirlArms, WideLegs, WiggleHips, WildArms, WildLegs (58)
MOT	CrowdAvoidance, InTheDark, LawnMower, OnHeels, OnPhoneLeft, OnPhoneRight, OnToesBentForward, OnToesCrouched, Rushed (9)
OBJ	DragLeftLeg, DragRightLeg, DuckFoot, Flapping, ShieldedLeft, ShieldedRight, Swimming, SwingArmsRound, SwingShoulders (9)

A.4 Inference times

To evaluate the inference efficiency of our submodules, full model, and baseline methods for stylized text2motion tasks, we report the average Inference Time per Sentence measured in seconds (AITS) [6], in Table 5. The AITS is calculated by setting the batch size to 1 and excluding the time cost for model and dataset loading on an NVIDIA A5000 GPU.

A.5 More details on classifier-based style guidance

In our experiments, we observed a phenomenon similar to that described in Text2Image [59]: In the early denoising stages, the generated motion gradually transitions from random movement to motion that adheres to the content text. Once the global motion content is shaped, subsequent denoising stages primarily

Sub-Modules	MLD	w/o adaptor	w/o classifier-based	Methods *Overall*	Ours	MLD + Motion Puzzle	MLD + Aberman et al.
Time (s)	0.2139	2.5081	0.5563	Time (s)	3.1133	0.2420	0.2275

Table 5: Inference time. We report the Average Inference Time per Sentence (AITS) in seconds for baselines and each submodule of ours on stylized text2motion tasks.

focus on modifying the local details and enhancing the quality of the motion. Introducing classifier-based style guidance at an early stage not only poses challenges in steering the motion toward the desired style but also affects the motion's adherence to the content text. Therefore, we apply classifier-based style guidance near the last stage, once the rough outline of the global motion content has been established and the focus shifts to modifying local details. Moreover, we can iterate classifier-based guidance multiple times K to improve the steered accuracy:

$$K = \begin{cases} K_e & \text{if } T_s < t < T, \\ K_l & \text{if } t \leq T_s. \end{cases}$$

We use $K_e = 0$, $K_l = 5$, and $T_s = 300$ in our experiments.

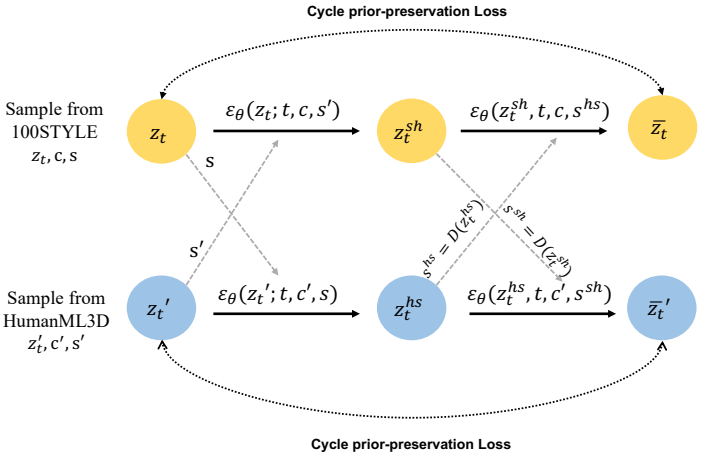

Fig. 8: Visual pipeline of the cycle prior-preservation loss.

A.6 More details on cycle prior-preservation loss

We introduce the cycle prior-preservation loss to ensure that generated motion retains content-invariant characteristics from the content text. Fig. 8 illustrates

the cycle prior-preservation loss's visual pipeline. At timestep t, the process begins with sampling content text $\boldsymbol{c}$, style motion sequence $\boldsymbol{s}$, and noisy motion latent $\boldsymbol{z}_t$ from the 100STYLE dataset, alongside their equivalents $\boldsymbol{c}'$, $\boldsymbol{s}'$, and $\boldsymbol{z}'_t$ from the HumanML3D dataset. Following this, we facilitate the transfer of content and style conditions between these datasets, yielding z_t^{sh} and z_t^{hs}. Decoding z_t^{hs} into the motion space generates the s^{hs} motion sequence. Viewed as a style motion sequence, s^{hs} is combined with the original content text $\boldsymbol{c}$ to reconstruct the noisy latent $\bar{\boldsymbol{z}}_t$. The cycle prior-preservation loss then operates between the original noisy latent $\boldsymbol{z}_t$ and the reconstructed noisy latent $\bar{\boldsymbol{z}}_t$.

A.7 User study details

To mitigate the potential challenges in participant selection when they are asked to rank or score various methods, we developed an online questionnaire with pairwise A/B tests. We randomly selected 12 sets of stylized motion for the stylized text-to-motion task and 10 sets for the motion style transfer tasks. We recruited 22 human subjects from various universities, representing a range of academic backgrounds, to participate in our study. At the start of the user study, we introduced the concept of motion stylization, providing examples of both the content text/motion and style motion for reference. With the reference style motion and content text/motion provided, participants were asked to evaluate and choose the better one based on the dimensions of Realism, Style Reflection, and Content Preservation, respectively. As shown in Fig. 7, our approach achieves better performance than the baselines on two tasks across three evaluation dimensions.

A.8 More ablation studies

Varying the weight of Classifier-based style guidance. Due to the flexibility of the style guidance weights, we explore the effects of varying the classifier-based style guidance weight in Fig. 9. We observe that increasing the classifier-based style guidance weight boosts the SRA metric but reduces R Precision, MM Dist, and FID, which means less content preservation but reflecting style more accurately. It is observed that when the absolute value of the weight of classifier-based style guidance τ exceeds 0.2, the rate of increase for SRA metrics slows down, yet the other metrics continue to deteriorate rapidly. Therefore, we set $\tau = -0.2$ as a trade-off.

Varying the weights of the classifier-free style guidance. Similar to how we can adjust the weights of classifier-based style guidance to balance style reflection and content preservation, as discussed in Sec. A.8, adjusting the weights of classifier-free style guidance also involves a trade-off. Fig. 10 illustrates the effects of varying the classifier-free style guidance weights w_s, while setting $\tau = 0$. As the weights w_s increase, the SRA gradually increases, while the R-precision and FID metrics deteriorate. It is observed that when w_s exceeds 1.5, FID, R

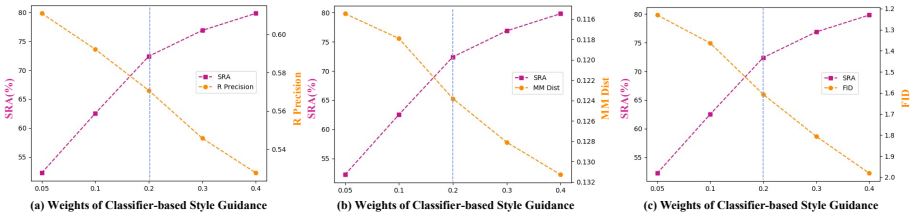

Fig. 9: Varying the weights of the classifier-based style guidance.

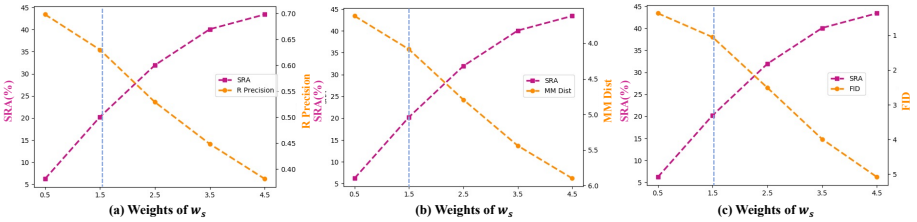

Fig. 10: Varying the weights of the classifier-free style guidance.

Precision, and MM Dist decrease more rapidly, whereas SRA continues to increase at the same rate. Therefore, we set $w_s = 1.5$ to prevent rapid deterioration in content preservation metrics while ensuring optimal performance in the SRA metric.

The alternative approach of prior preservation loss. In Sec. 3.3, we introduce our prior preservation loss, which involves sampling instances from the HumanML3D dataset as well as from the 100STYLE dataset, and then calculating the loss to prevent 'content-forgetting.' A straightforward alternative approach involves simply combining the 100STYLE and HumanML3D datasets to create a larger dataset, and then only utilizing L_{std} to fine-tune the style adaptor. Given the larger number of samples in the HumanML3D dataset compared to the 100STYLE dataset, this approach struggles to effectively capture style features from instances in the 100STYLE dataset and maintain learned content in a single optimization step. We term this alternative method the *combined dataset* approach, utilizing it to train the style adaptor across the same number of training iterations. Compared to the second and third rows in Table 6, the *combined dataset* approach shows markedly worse performance in content preservation metrics, such as FID and MM Dist values, indicating a failure to preserve content. These results demonstrate that our simple prior preservation loss can effectively learn style features and simultaneously preserve the learned content with minimal training steps.

Table 6: Ablation Studies on HumanML3D Content and 100STYLE Styles.

Method	FID↓	Foot skating ratio↓	MM Dist↓	R-precision↑ (Top-3)	Diversity→	SRA(%)↑
Ours (on all)	1.609	0.124	4.477	0.571	9.235	72.418
combined dataset	3.892	0.332	6.152	0.379	6.833	57.573

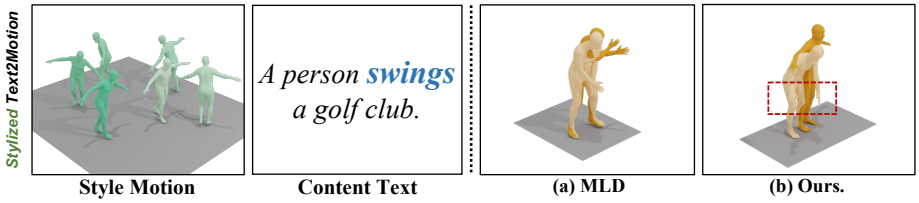

Fig. 11: A visual example showing conflicts between content text and style motion in a specific body part.

A.9 Limitation and future plans

A primary limitation of our approach is its reliance on a pre-trained motion diffusion model, which impacts the realism of the generated motions. Consequently, our approach may produce motions with foot skating for certain content texts. We present these failure cases in the supplementary video. Incorporating realism guidance [53] or physical constraints [60] might be a promising direction to improve the realism of the generated motions.

Another limitation is that, due to the classifier-based style guidance potentially requiring iteration, our approach is more time-consuming than MLD by nearly 10 times. A potential direction for improvement involves decreasing the number of denoising steps, inherently reducing the iterations required for classifier-based guidance. Exploring the integration of a one-step model, such as the consistency model [44], in the motion generation could be a valuable direction.

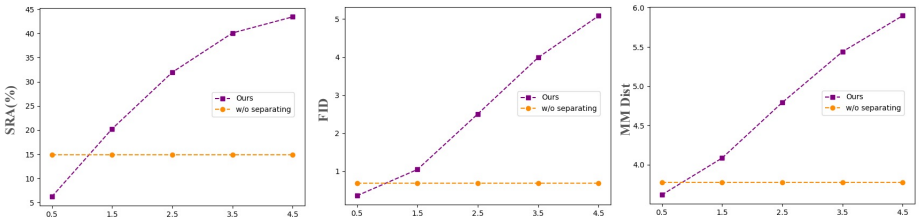

Fig. 12: Comparing our approach with the variant without separating the classifier-free style guidance from content guidance.