IMPROVE TEMPORAL CONSISTENCY IN DIFFUSION MODELS THROUGH NOISE CORRELATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have emerged as a powerful tool for generating diverse types of data, including sequential data such as audio, video, and motion. As the temporal consistency in sequential data is crucial for maintaining fidelity and realism, this paper introduce the **A**uto**R**egressive **T**emporal diffusion (ARTDiff) approach to address the challenge of temporal consistency in diffusion models. ARTDiff offers a straightforward and efficient solution that requires minimal computational overhead. Our proposed ARTDiff method leverages the inherent autoregressive dependence structure in time by introducing a Gaussian noise distribution whose correlations between time frames have a functional form in terms of time difference. This design explicitly captures the temporal dependencies and enhances the consistency in generated sequences. We evaluate the effectiveness of ARTDiff on audio, motion and video generation tasks. Experimental results demonstrate that ARTDiff significantly improves the fidelity and realism of generated samples compared to baseline diffusion models. The simplicity and efficiency of ARTDiff make it a practical choice for incorporating temporal consistency in diffusion-based generation models.

1 INTRODUCTION

Deep generative models have found applications across a wide array of data types, including images, audio, video, motion, and more. Among this diverse range, a significant subset is sequential data. Unlike other data types whose order does not matter, sequential data possess a temporal dimension that is distinct from other feature dimensions. The coherence and dynamics of this particular dimension play a vital role in generating valid samples. To tackle this challenge, an intuitive solution is to employ an autoregressive framework, where each subsequent data frame is generated by conditioning explicitly on previous frames. Notable examples of such frameworks include Convolutional Neural Networks (CNN) (Van den Oord et al., 2016a;b), Transformers (Radford et al., 2018; Brown et al., 2020), autoregressive Generative Adversarial Networks (GANs) (Morrison et al., 2021), and autoregressive normalizing flows (Valle-Pérez et al., 2021). The autoregressive framework establishes correlations among time frames; yet it may lead to an excessively slow sampling time, especially for long sequences.

On the other hand, there is a surge of interest in the family of diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020). They have been extensively used in generating sequential data such as audio (Kong et al., 2021; Yang et al., 2023), video (Ho et al., 2022; Blattmann et al., 2023), and motion (Tevet et al., 2022; Chen et al., 2023). These models have garnered recognition due to their stable training and efficient parallel sampling techniques, which can be attributed to their non-autoregressive nature that circumvents the need for sequential sampling. However, the non-autoregressive property of diffusion models raises the question of their ability to effectively capture and preserve the temporal dependencies inherent in sequential data.

Various approaches have been explored to incorporate the autoregressive framework into diffusion models, with the goal of enhancing temporal consistency. For instance, Han et al. (2023) explicitly introduced the autoregressive framework by iteratively generating motion clips conditional on the previous timestamp. Rasul et al. (2021) added a Recurrent Neural Network (RNN) alongside to model the sequence, where the reverse diffusion process at each time frame t is conditioned on the t-th output of the RNN. Unsurprisingly, these methods inherit the potential drawback of the autoregressive framework, as the time-consuming reverse diffusion process needs to be repeated multiple times. Alternatively, there are more subtle designs that introduce autoregressiveness. For example, in Luo

et al. (2023), the video sequences are divided into distinct frames, and instead of generating the entire video as a whole, the model generates the difference between each frame and a central frame. Latent diffusion models (Lovelace et al., 2022; Blattmann et al., 2023; Chen et al., 2023) utilized encoders and decoders to transform the target data into a latent space with reduced dimensions. The diffusion models then operate in this latent space whose temporal dynamics are concentrated. However, these methods either require complex training schemes or struggle to effectively capture long-range dependencies because of the limited latent space dimension.

This paper presents a novel and seamless approach to address temporal inconsistency by integrating autoregressiveness directly into the diffusion process. Specifically, we design a new sampling distribution where the temporal correlations are added into noises. These correlations adhere to an autoregressive structure, with stronger correlations among closer time frames and weaker correlations among farther time frames. Inspired by autocorrelation patterns of autoregressive (AR) models in time series literature (Fuller, 1996; Cryer & Chan, 2008) and temporal patterns of Recurrent Neural Networks (RNNs) (Huang et al., 2023) and state space models (Gu et al., 2022; Smith et al., 2023), we propose three different correlation designs. A localized design is also devised to adapt for extremely long sequences. Within the local window, the correlations decay exponentially as the distance between time frames increases. The noises for different windows also follow an order-1 autoregressive model to strengthen the temporal consistency. The implementation of our new sampling scheme is straightforward and incurs nearly no computational overhead as empirically validated in Section 3.4.

The main contributions of our paper are threefold:

1. We propose a novel method to address the challenge of temporal inconsistency in diffusion models by introducing autoregressive techniques directly into the diffusion process. By designing a new sampling distribution that incorporates temporal correlations into the noise, we enable the diffusion models to effectively capture and preserve the temporal dependencies inherent in sequential data.
2. The new method maintains the efficient non-autoregressive sampling scheme of diffusion models while incurring minimal computational overhead. This allows the generation of long sequences in an efficient and scalable manner, making our approach applicable to real-world scenarios that involve the synthesis of large-scale sequential data.
3. We extensively evaluate the effectiveness of our method in enhancing temporal coherence, preserving fidelity, and capturing long-range dependencies through various sequential data generation tasks, including audio, motion and video synthesis. Moreover, our ablation study reveals an interesting finding that highlights the potential harm caused by a mismatch in sampling distribution, particularly for long sequences.

1.1 Other related works

Noise design in diffusion models Several works have explored the design of noises used in diffusion models, each with specific objectives. Nichol & Dhariwal (2021) and Chen (2023) investigated alternative noise scheduling methods beyond the linear scheduling in Ho et al. (2020). Avrahami et al. (2022) blended the noised background and foreground images in each diffusion step to enable local editing, and similar idea is also adopted in Couairon et al. (2023). Regarding the noise distribution, Nachmani et al. (2021) studied the use of Gamma distribution and a mixture of Gaussian distributions, which offer more degrees of freedom. Luo et al. (2023) separated the noise into two parts, one for the central frame and another for the difference from the central frame to improve temporal consistency in video generation. Ge et al. (2023) directly imposed an AR(1) model to the noises, which does not account for other temporal decay patterns and long-term dependencies; instead, our correlation design has richer and more flexible patterns. Lee et al. (2022) introduced an informative prior distribution for noise by leveraging conditional context information. However, their method requires manual design adjustments for each new data type and task. Our paper also considers modifying the noise distribution, however, our approach can be easily implemented in both conditional and unconditional sequential generation tasks and be generalized for different data types.

Autoregressive framework in sequence modeling In the context of sequence modeling, capturing the temporal dynamics with autoregressive frameworks without compromising computational efficiency is also of interest. Recent advancements include integrating RNNs into Transformers while maintaining parallelization (Huang et al., 2023) and using state space models to capture temporal dependencies

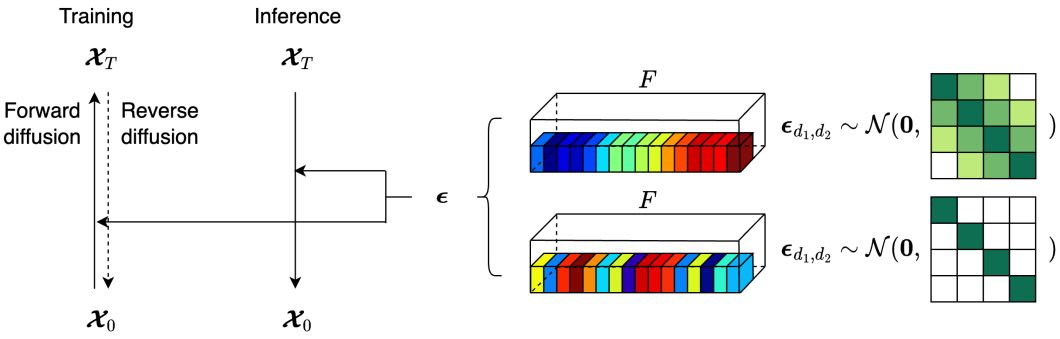

Figure 1: Illustration of the comparison between ARTDiff and vanilla diffusion. In the upper part, ARTDiff utilizes Gaussian noises that are generated from designed distributions featuring temporal correlations. In contrast, noises are generated from standard Gaussian in vanilla diffusion.

(Gu et al., 2022; Smith et al., 2023). Despite the different autoregressive frameworks utilized in these studies, they all highlight that the temporal dynamics of RNNs and state space models can be characterized by two fundamental patterns: exponential decays and damped sine or cosine waves. Remarkably, these patterns also match the shapes of autocorrelation functions for AR models in time series literature (Fuller, 1996; Cryer & Chan, 2008).

2 METHODS

2.1 REVISIT DIFFUSION MODELS

We start with reviewing the vanilla diffusion process following the DDPM in Ho et al. (2020) and consider a case where the data is sequential. Denote a sequential data sample with the number of time frames (or sequence length) equal to F by $\mathcal{X} \in \mathbb{R}^{D_1 \times \cdots \times D_p \times F}$, where the last dimension corresponds to the temporal dimension and the first p dimensions are other feature dimensions.

In vanilla diffusion process, the $p+1$ dimensions are not distinguished and we may treat the tensor $\mathcal{X}$ as a vector $\boldsymbol{x}$. The diffusion models encompass two processes: forward diffusion and reverse diffusion. In the forward diffusion process, the sample $\boldsymbol{x}$ is gradually destroyed by adding standard Gaussian noises at each diffusion step $t+1$:

$$\boldsymbol{x}_{t+1} = \sqrt{1-\beta_t}\boldsymbol{x}_t + \sqrt{\beta_t}\boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}),$$

where $\boldsymbol{I} \in \mathbb{R}^{D_1 \cdots D_p F \times D_1 \cdots D_p F}$. Conversely, the reverse diffusion process initiates with a standard Gaussian noise and progressively removes the noise to obtain a high-quality sample. Notably, a neural network is employed to learn the distribution of $\boldsymbol{x}_{t-1}$ given $\boldsymbol{x}_t$:

$$\boldsymbol{x}_{t-1} | \boldsymbol{x}_t \sim \mathcal{N}\left(\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)\right).$$

In the inference step, in addition to the neural denoising procedure, another standard Gaussian noise is added back to the sample in order to increase diversity.

2.2 AUTOREGRESSIVENESS IN TEMPORAL NOISE

Manipulating the noise within diffusion models has proven to be an effective approach for enhancing generation quality, as discussed in Section 1.1. In this paper, we propose a noise design aimed at improving the temporal consistency of generated data. Instead of using standard Gaussian noise during both training and inference stages, we introduce autoregressive-structured correlations for the noises across time. This targeted approach enables a more cohesive and coherent generation process.

To begin with, consider a simple case where the data sample $\boldsymbol{x} \in \mathbb{R}^F$, i.e. it is a univariate time sequence. Let the noise $\boldsymbol{\epsilon} \in \mathbb{R}^F$ follow a Gaussian distribution with mean zero and covariance matrix $\boldsymbol{\Sigma}$ having a parametric form: the (i,j)-th entry $\boldsymbol{\Sigma}_{i,j} = f_{|i-j|}(\boldsymbol{\lambda})$ where $\boldsymbol{\lambda}$ is the hyper-parameter. The structure of $\boldsymbol{\Sigma}$ is artificially designed so that the $\boldsymbol{\epsilon}$'s and hence the noised $\boldsymbol{x}_t$'s in the diffusion process exhibit nonzero correlations that depend on their temporal distance. Moreover, it is natural to

assume these correlations decay as the temporal distance of two time frames increases. We therefore provide three choices for $f_{|i-j|}(\boldsymbol{\lambda})$, which match the common temporal decay patterns provided by the simple yet powerful autoregressive (AR) model in time series literature (Box et al., 2008; Basu & Michailidis, 2015).

The first option is to set $f_{|i-j|}(\boldsymbol{\lambda}) = \lambda^{|i-j|}$ with $|\lambda| \in (0,1)$ and therefore

$$\boldsymbol{\Sigma}(\lambda, F) = \begin{pmatrix} 1 & \lambda & \lambda^2 & \cdots & \lambda^{F-1} \\ \lambda & 1 & \lambda & \cdots & \lambda^{F-2} \\ \lambda^2 & \lambda & 1 & \cdots & \lambda^{F-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda^{F-1} & \lambda^{F-2} & \lambda^{F-3} & \cdots & 1 \end{pmatrix}. \tag{1}$$

This resembles the AR(1) structure in time series, where the correlation between adjacent time frames is the strongest while decaying exponentially with increasing temporal distance. Alternatively, we may let $\boldsymbol{\lambda} = (\gamma, \theta)$ with $\gamma \in (0,1)$ and $\theta \in (-\pi/2, \pi/2)$ and set $f_{|i-j|}(\boldsymbol{\lambda}) = f_{|i-j|}(\gamma, \theta) = \gamma^{|i-j|} \cos\big(|i-j|\theta\big)$ or $\gamma^{|i-j|} \sin\big(|i-j|\theta\big)$. Such two designs of $\boldsymbol{\Sigma}$ render the temporal decay pattern to be exponentially damped cosine or sine waves and mimics the autocorrelation structures of an AR model when the corresponding AR characteristic polynomial equation has complex roots; see Section 2.6 of Fuller (1996) and Chapter 4.3 of Cryer & Chan (2008) for details.

However, in certain applications, the temporal dimension can be excessively long, making it impractical to sample from a full $F \times F$ covariance matrix. In such scenarios, we adopt a localized approach by introducing correlations only within local windows. Specifically, we let $\boldsymbol{\epsilon} = (\boldsymbol{\epsilon}_1^\top, \boldsymbol{\epsilon}_2^\top, \cdots, \boldsymbol{\epsilon}_n^\top)^\top$ with $\boldsymbol{\epsilon}_i \in \mathbb{R}^w$, where n is the number of local windows, w is the window size and $F = nw$. Then we let $\boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma}(\lambda, w))$. To further enhance the consistency between local windows, we can model each noise clip $\boldsymbol{\epsilon}_i$ using an AR(1) model:

$$\boldsymbol{\epsilon}_{i+1} = \sqrt{c}\boldsymbol{\epsilon}_i + \sqrt{1-c}\tilde{\boldsymbol{\epsilon}} \quad \text{with} \quad \boldsymbol{\epsilon}_1, \tilde{\boldsymbol{\epsilon}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}(\lambda, w)). \tag{2}$$

By this construction, the combined noise $\boldsymbol{\epsilon}$ will also have a closed-form distribution with the covariance matrix

$$\boldsymbol{\Sigma}(\lambda, c, w, n) = \begin{pmatrix} \boldsymbol{\Sigma}(\lambda, w) & \sqrt{c}\boldsymbol{\Sigma}(\lambda, w) & \cdots & (\sqrt{c})^{n-1}\boldsymbol{\Sigma}(\lambda, w)) \\ \sqrt{c}\boldsymbol{\Sigma}(\lambda, w) & \boldsymbol{\Sigma}(\lambda, w) & \cdots & (\sqrt{c})^{n-2}\boldsymbol{\Sigma}(\lambda, w) \\ \vdots & \vdots & \ddots & \vdots \\ (\sqrt{c})^{n-1}\boldsymbol{\Sigma}(\lambda, w) & (\sqrt{c})^{n-2}\boldsymbol{\Sigma}(\lambda, w) & \cdots & \boldsymbol{\Sigma}(\lambda, w) \end{pmatrix}. \tag{3}$$

It is worth noting that the overall covariance matrix exhibits similar exponential-type decays of local covariance structures due to the AR construction.

To extend to the multivariate case where $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1 \times \cdots \times D_p \times F}$, we can adopt a simple and straightforward approach by assuming independence in the first p dimensions while introducing correlations solely in the temporal dimension. Specifically, let $\boldsymbol{x}_{d_1, \cdots, d_p} \in \mathbb{R}^F$ denote a vector extracted from $\boldsymbol{\mathcal{X}}$ and let the corresponding $\boldsymbol{\epsilon}_{d_1, \cdots, d_p}$ follow the aforementioned Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Please refer to Figure 1 for an illustration. Although the independent and identically distributed setting suffices, it can be extended to incorporate correlations in spatial dimension as well. For more comprehensive information, please see Section 5 and the discussions therein.

In addition, changing the covariance matrix from identity to a general symmetric positive semi-definite matrix will not affect the DDPM framework and can be implemented very easily, see Appendix A.2 and B.1 for more details. To summarize, we present the complete autoregressive temporal diffusion (ARTDiff) algorithm for training and inference in Algorithms 1 and 2.

Algorithm 1 ARTDiff Training

Given λ, c, w, determine $\boldsymbol{\Sigma}$
repeat
  $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$
  $t \sim$ Uniform$(\{1, \ldots, T\})$
  $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$
  Take gradient descent step on $\nabla_\theta ||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)||^2$
until converged

Algorithm 2 ARTDiff Inference

Given $\boldsymbol{\Sigma}$ same as in training
$\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $t = T, \ldots, 1$ **do**
  $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ if $t \geq 1$, else $\boldsymbol{z} = \mathbf{0}$
  $\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)) + \sigma_t \boldsymbol{z}$
end for
return $\boldsymbol{x}_0$

3 EXPERIMENT

To show that the proposed ARTDiff can improve the fidelity of generated samples, we conduct experiments on three types of sequential data, namely audio, motion and video, whose generation inherently requires temporal consistency. We compare ARTDiff's performances with vanilla diffusion. All experiments are performed on a single A100 GPU.

3.1 AUDIO GENERATION

Audio data feature a large temporal dimension compared to other data types that generative models often work on, which brings in challenge in modelling and capturing the correlations in such long sequences (Mehri et al., 2017; Kreuk et al., 2023). To show that the ARTDiff can help alleviate this problem and improve the quality of generation, we perform two tasks on audio data: neural vocoding and unconditional generation, and compare the performance of ARTDiff with the vanilla diffusion. We adopt two baseline models: DiffWave (Kong et al., 2021) and DiffWave-Sashimi (Goel et al., 2022), and follow the same training scheme in Goel et al. (2022).

For neural vocoding, we utilize the LJSpeech (Ito & Johnson, 2017) dataset. It is an open-access repository comprising 13,100 concise audio clips ranging from 1 to 10 seconds in length, with a total length of approximately 24 hours. During training, the sequence length is $F = 16,000$ for each batch; and this length is extended to that of a complete audio clip with a sampling rate of 22,050Hz during inference, making F to range from $\sim$20k to $\sim$200k. To evaluate the performance, we generate 13,100 samples condition on the true mel-spectrogram of original audio clips and adopt the KL-divergence (KL), Fréchet Inception Distance (FID) in Heusel et al. (2017) and Fréchet audio distance (FAD) in Kilgour et al. (2019)[1].

For unconditional generation task, we employ the SC09 (Donahue et al., 2019) dataset, a subset of the comprehensive Speech Commands (Warden, 2018) dataset. SC09 is a challenging benchmark for unconditional speech generation, featuring 1-second clips of utterances of the digits zero through nine, recorded by various speakers with diverse accents and noise conditions. The dataset has a sampling rate of 16kHz, i.e. the number of time frames in data is $F = 16,000$. To assess the generative quality, we randomly generate 1,024 samples from each model and compare the three metrics: FID, FAD, and Inception score (IS) in Salimans et al. (2016).

Table 1: Reported metrics for DiffWave versus DiffWave + ARTDiff, and DiffWave-Sashimi versus DiffWave-Sashimi + ARTDiff on two audio generation tasks: (a) neural vocoding and (b) unconditional generation.

(a) Neural vocoding

Model	KL	FID	FAD
DiffWave	0.023	1.705	4.199
+ARTDiff	**0.018**	**0.893**	**2.656**
DiffWave-sashimi	0.056	4.456	5.933
+ARTDiff	**0.016**	**1.171**	**2.564**

(b) Unconditional generation

Model	FID	FAD	IS (std)
DiffWave	5.229	2.057	**1.354** (0.104)
+ARTDiff	**3.078**	**1.276**	1.242 (0.044)
DiffWave-sashimi	**2.955**	2.264	1.078 (0.019)
+ARTDiff	3.910	**1.579**	**1.236** (0.065)

The results for neural vocoding and unconditional generation are gathered in Table 1 (a) and (b) respectively. We can see that by changing to ARTDiff, the performance of audio synthesis is greatly improved, especially in the neural vocoding task and in the FAD (a tailor-made metric to evaluate

[1]The evaluation methods follow the guidelines in Liu et al. (2023)(https://github.com/haoheliu/audioldm_eval)

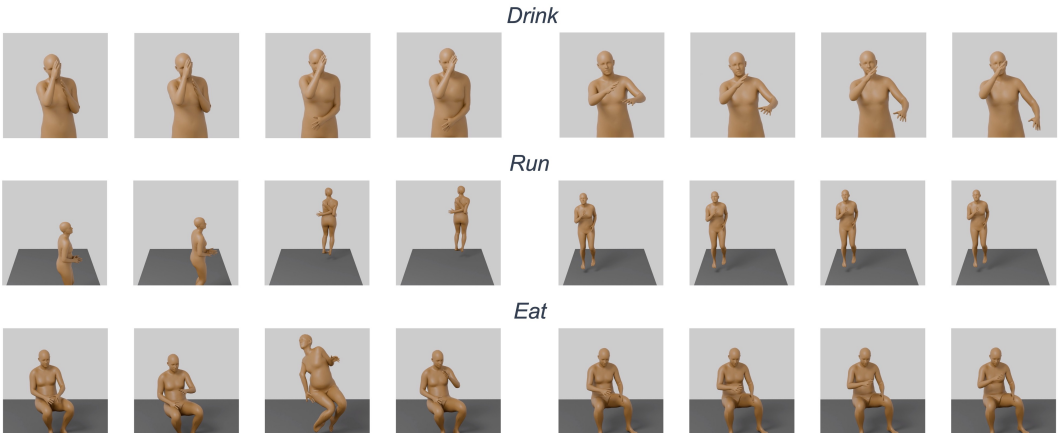

Figure 2: Visualizations of generated motion frames in action-to-motion task. Motion clips in the left column are produced by MDM and those in the right column are produced by MDM + ARTDiff.

generated audio) of the unconditional generation task, showing that the ARTDiff is very flexible and can be extended to very long sequences. The inconsistencies observed in IS and FID scores presented in Table 1(b) may be caused by insufficient sample size and FID's focus on evaluating image generation, respectively.

3.2 Human Motion Generation

Human motion is another type of sequential data where temporal consistency is crucial. We thus conduct an extensive investigation into the effectiveness of ARTDiff across three tasks: unconditional generation, action-to-motion generation, and text-to-motion generation. Throughout all three tasks, we utilize the MDM (Tevet et al., 2023) as the backbone and replace the original diffusion mechanism with the proposed temporal consistency diffusion, while maintaining the training scheme outlined in the original paper. The reported results for each model are based on the checkpoint with the best evaluation performance during training.

Unconditional generation We employ the HumanAct12 dataset (Guo et al., 2020) that is originally designed for the action-to-motion task. In this case, we remove the action annotations to adapt the dataset for unconditional purposes.

Action-to-motion generation The aforementioned HumanAct12 dataset are adopted, which consists of 12 action classes, with varying sample sizes per class ranging from 47 to 218. The sequence length for HuamanAct12 is set to 60 in both tasks.

Text-to-motion generation For this task, we utilize two datasets: HumanML3D (Guo et al., 2022) and KIT (Plappert et al., 2016). The former consists of 14,616 samples, while the latter contains 3,911 samples. The datasets are annotated with 44,970 and 6,278 textual descriptions, respectively.

Temporal consistency measure To quantitatively assess the temporal consistency of the generated data, we introduce a novel metric called Temporal Consistency Metric (TCM). The underlying notion is that, at every time point, the data frames should exhibit smooth transitions without any outliers. Inspired by the approach presented in Lai et al. (2018), we define the TCM as

$$\text{TCM} = \frac{1}{F}\sum_{t=1}^{F-1} \|d_{t+1} - d_t\|_2,$$

where d_t is the data frame at time t in vectorized form. It is important to note that the TCM will yield a value of zero if the generated sample is static. Consequently, it cannot serve as the sole criterion for evaluating the performance of a generative model. Instead, this metric should be considered in conjunction with an overall assessment of the sample quality.

In the four tasks performed, as presented in Table 2 (a)-(c), the ARTDiff can improve the quality of synthesised motion further in most metrics, and the temporal consistency is also enhanced as evidenced by the TCM measure. We also present visualizations of the action-to-motion generation

task in Figure 2, using the MDM and ARTDiff for comparison. The figure includes three example prompts: "drink", "run", and "eat". It can be observed that the motion generated by the baseline model lacks temporal consistency, as evidenced by the movement of the left arm in the "drink" prompt and the inclusion of irrelevant frames in the "eat" prompt. In contrast, our model generates motions and postures with a significantly higher level of smoothness and temporal consistency.

Table 2: Reported metrics for MDM and MDM+ARTDiff on three motion generation tasks: (a) unconditional generation; (b) action-to-motion generation; (c) text-to-motion generation tasks. Best results are highlighted in boldface.

(a) Unconditional generation

Model	FID	KID	Precision	Recall	Diversity	TCM
MDM	32.82	0.378	0.695	0.638	**17.48**	0.063
+ARTDiff	**30.97**	**0.377**	**0.701**	**0.658**	17.04	**0.060**

(b) Action-to-motion generation on Humanact12 dataset.

Model	Accuracy	Diversity	FID	Multimodality	TCM
MDM	0.989	6.852	0.100	**2.515**	0.488
+ARTDiff	**0.991**	**6.880**	**0.097**	2.426	**0.466**

(c) Text-to-motion generation on HumanML3D and KIT dataset.

Model	Top-3 Accuracy	Diversity	FID	Multimodality	Matching Score	TCM
HumanML3D						
MDM	**0.640**	9.607	0.4775	**2.697**	5.3096	**11.562**
+ARTDiff	0.639	**9.7052**	**0.3797**	2.650	**5.312**	11.760
KIT						
MDM	**0.405**	**11.016**	0.416	2.042	9.939	24.725
+ARTDiff	0.396	10.863	**0.401**	**2.035**	**9.314**	**20.709**

3.3 VIDEO GENERATION

Video generation is a rapidly involving topic in neural generative models, and most of the existing works leverage the well-developed image generation models, which generate high-quality frames but introduce the inconsistency problem along the temporal direction. Therefore, our ARTDiff may be a promising solution to address the frame inconsistency in video generation tasks. We take the one-shot text-to-video generation model Tune-A-Video (Wu et al., 2023) as the baseline. It consists of a fine-tuning stage in which the pretrained image generation models (e.g. stable diffusion) are fine-tuned on a single reference video-caption pair, and an inference stage in which new videos are generated conditioned on edited prompts.

To improve temporal consistency, we add our ARTDiff in both the fine-tuning and inference stages. In particular, ARTDiff is added to the DDIM inversion to produce a correlated initial noise. The reference videos are sourced from the sample videos in Wu et al. (2023) and the DAVIS dataset (Pont-Tuset et al., 2017). The corresponding captions are generated by the BLIP-2 model (Li et al., 2023). Details of the dataset and prompts can be found in Appendix B.3. To evaluate the quality of generated videos, we adopt three quantitative metrics: CLIP score, CLIP similarity and pixel TCM in Table 3. The CLIP score measures whether the video matches the corresponding edited prompt and is averaged over all frames. The CLIP similarity is the average cosine similarity between the CLIP image embeddings of adjacent frames. In addition, the pixel TCM focuses more on the pixel level and can better capture the abrupt change in video content. Qualitatively, we visualize the generated video samples in Figure 3, which showcase the enhanced temporal consistency brought by ARTDiff.

Table 3: Reported metrics for Tune-A-Video versus Tune-A-Video + ARTDiff on one-shot text-to-video generation task.

Model	CLIP score	CLIP similarity	Pixel TCM
Tune-A-Video	33.056	0.972	0.048
+ARTDiff	**33.362**	**0.974**	**0.040**

Figure 3: Visualizations of generated video frames. Video clips in the left column is from Tune-A-Video and those in the right column are produced by Tune-A-Video + ARTDiff.

3.4 Computational time efficiency

A notable concern when transitioning from a standard Gaussian noise distribution to a general Gaussian distribution is the potential increase in computational time during the random sampling phase. To address this concern, we choose two tasks: unconditional audio generation with the number of time frames being $F = 16,000$ and one-shot text-to-video generation with $F = 24$, and measure the training and inference time for vanilla diffusion and ARTDiff. The comparison results are presented in Figure 4. In terms of training time in the audio task, the additional amount attributed to ARTDiff is not significant. When F is smaller, the discrepancy in time will be further reduced or even negligible as shown in the right panel.

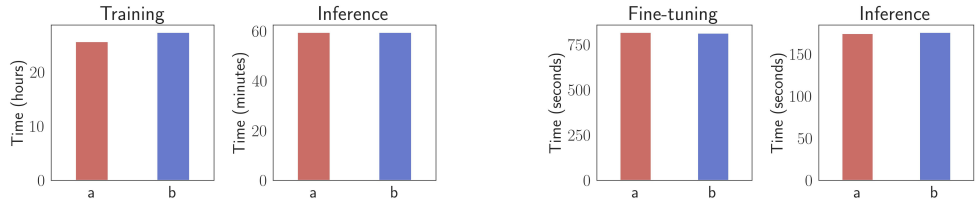

Figure 4: Time comparison of case (a) vanilla diffusion and (b) ARTDiff, in unconditional audio generation task (left panel) and one-shot text-to-video generation task (right panel). In the audio task, the training time is recorded for 100k iterations and the inference time is for generating 1,024 samples with batch size 128. In the video task, the finetuning is conducted for 500 steps and the inference time contains DDIM inversion and the generation of four videos based on edited prompts.

4 ABLATION STUDY

The proposed ARTDiff approach incorporates temporally correlated noises during both training and inference stages. However, it is also possible to introduce the correlated noises during inference only (Zhu et al., 2023). To demonstrate the benefits and, at times, the necessity of maintaining a consistent distribution between training and diffusion, we conduct an ablation study. The study compares the full ARTDiff approach, where correlated noises are used in both training and inference, with a configuration employing vanilla diffusion during training and ARTDiff during inference.

To evaluate these two settings, we employ the SC09 dataset for unconditional audio generation and the Humanact12 dataset for action-to-motion generation. In the case of unconditional audio generation, we utilize the checkpoint at 500k iterations, trained using vanilla diffusion, and generate 1,024 samples following Algorithm 2, with the noises generated from the same distribution as ARTDiff. For the action-to-motion generation task, the inference checkpoint selected is at 450k interations, which is the best checkpoint obtained during training.

Based on the results presented in Table 4, it is evident that DiffWave + ARTDiff (inference) exhibits poor performance in the audio task. However, in the motion task, this configuration demonstrates performance levels between MDM and MDM + ARTDiff. This discrepancy can be attributed to the lengthy sequence length of $F = 16,000$ in audio data, where a distribution shift can lead to a mismatch between the trained denoising model and the noises used during inference. Consequently, it is crucial to ensure a closer alignment between the distribution employed during training and the distribution utilized in inference.

Table 4: Ablation study to compare backbone models + ARTDiff in both training and inference, with ARTDiff in inference only. The two conducted tasks are (a) unconditional audio generation and (b) action-to-motion generation on Humanact12 dataset.

(a) Unconditional audio generation

Model	FID	FAD	IS (std)
DiffWave	5.229	2.057	**1.354** (0.104)
+ARTDiff	**3.078**	**1.276**	1.242 (0.044)
+ARTDiff (inference)	86.670	24.139	1.100 (0.007)

(b) Action-to-motion generation on Humanact12 dataset

Model	Accuracy	Diversity	FID	Multimodality	TCM
MDM	0.989	6.852	0.100	2.515	0.488
+ARTDiff	**0.991**	**6.880**	**0.097**	2.426	**0.466**
+ARTDiff (inference)	0.987	6.866	0.102	**2.525**	0.488

5 CONCLUSION

By changing the noise distribution in diffusion models, we propose a new diffusion training and inference scheme called ARTDiff to improve the temporal consistency in sequential data generation. Specifically, we consider a Gaussian distribution where the correlations between two time frames exponentially decay with the time difference. This inherently introduces an autoregressive dependence structure in time dimension while maintaining the non-autoregressive nature of diffusion models. In addition, for long sequence tasks, a local window correlation scheme is proposed to alleviate the computational burden. Experiments on audio, motion and video generation tasks show the improvement of fidelity in generated samples using ARTDiff.

This paper can be extended along three directions below. First, while the correlation pattern of Gaussian noises is determined by several hyper-parameters, we may also consider more flexible, or even data-driven patterns where correlations are learned from the data itself. Secondly, the ARTDiff can be further applied to other sequential data such as time series and facial expressions. Finally, it is of interest to explore a more general dependence pattern in the spatial dimension. For example, one may consider a stronger dependence for the static background and a weaker dependence for the moving object.

REFERENCES

Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18208–18218, 2022.

Sumanta Basu and George Michailidis. Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, 43(4), aug 2015. doi: 10.1214/15-aos1315. URL https://doi.org/10.1214%2F15-aos1315.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22563–22575, 2023.

George E. P. Box, Gwilym M. Jenkins, and Gregory C. Reinsel. *Time series analysis: forecasting and control*. Wiley series in probability and statistics. John Wiley, Hoboken, N.J, 4th ed. edition, 2008. ISBN 9780470272848.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023.

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18000–18010, 2023.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=3lge0p5o-M-.

J.D. Cryer and K.S. Chan. *Time Series Analysis: With Applications in R*. Springer Texts in Statistics. Springer New York, 2008. ISBN 9780387759593. URL https://books.google.com.hk/books?id=bHke2k-QYP4C.

Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. In *ICLR*, 2019.

Wayne A. Fuller. *Introduction to statistical time series*. Wiley series in probability and statistics. J. Wiley, New York, 2nd ed. edition, 1996. ISBN 1-282-30767-3.

Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22930–22941, 2023.

Karan Goel, Albert Gu, Chris Donahue, and Christopher Ré. It's raw! audio generation with state-space models. In *International Conference on Machine Learning*, pp. 7616–7633. PMLR, 2022.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uYLFoz1vlAC.

Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 2021–2029, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5152–5161, June 2022.

Bo Han, Hao Peng, Minjing Dong, Chang Xu, Yi Ren, Yixuan Shen, and Yuheng Li. Amd autoregressive motion diffusion. *arXiv preprint arXiv:2305.09381*, 2023.

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. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Feiqing Huang, Kexin Lu, Yuxi CAI, Zhen Qin, Yanwen Fang, Guangjian Tian, and Guodong Li. Encoding recurrence into transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=7YfHla7IxBJ.

Keith Ito and Linda Johnson. The lj speech dataset. https://keithito.com/LJ-Speech-Dataset/, 2017.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms. In *INTERSPEECH*, pp. 2350–2354, 2019.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=a-xFK8Ymz5J.

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CYK7RfcOzQ4.

Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 170–185, 2018.

Sanggil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_BNiN4IjC5.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Weinberger. Latent diffusion for language generation. *arXiv preprint arXiv:2212.09462*, 2022.

Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10209–10218, 2023.

Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. SampleRNN: An unconditional end-to-end neural audio generation model. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SkxKPDv5xl.

Max Morrison, Rithesh Kumar, Kundan Kumar, Prem Seetharaman, Aaron Courville, and Yoshua Bengio. Chunked autoregressive gan for conditional waveform synthesis. In *International Conference on Learning Representations*, 2021.

Eliya Nachmani, Robin San Roman, and Lior Wolf. Non gaussian denoising diffusion models. *arXiv preprint arXiv:2106.07582*, 2021.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Matthias Plappert, Christian Mandery, and Tamim Asfour. The KIT motion-language dataset. *Big Data*, 4(4):236–252, dec 2016. doi: 10.1089/big.2016.0028. URL http://dx.doi.org/10.1089/big.2016.0028.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pp. 8857–8868. PMLR, 2021.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Ai8Hw3AXqks.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. In *European Conference on Computer Vision*, pp. 358–374. Springer, 2022.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=SJ1kSyO2jwu.

Guillermo Valle-Pérez, Gustav Eje Henter, Jonas Beskow, Andre Holzapfel, Pierre-Yves Oudeyer, and Simon Alexanderson. Transflower: probabilistic autoregressive dance generation with multimodal attention. *ACM Transactions on Graphics (TOG)*, 40(6):1–14, 2021.

Aäron Van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: Agenerative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016a.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016b.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *ArXiv*, abs/1804.03209, 2018.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7623–7633, 2023.

Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

Lingting Zhu, Xian Liu, Xuanyu Liu, Rui Qian, Ziwei Liu, and Lequan Yu. Taming diffusion models for audio-driven co-speech gesture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10544–10553, 2023.

APPENDIX

This Appendix contains three sections. The first section provides theoretical derivations for the covariance matrix structure and DDPM framework. The second section illustrates the ARTDiff formulation in images. An efficient method to construct the newly designed covariance matrix and additional implementation details are included in the third section.

A THEORETICAL DERIVATIONS

A.1 PROOF OF EQUATION 3

Proposition 1. *Given* $\boldsymbol{\epsilon} = (\boldsymbol{\epsilon}_1^\top, \boldsymbol{\epsilon}_2^\top, \cdots, \boldsymbol{\epsilon}_n^\top)^\top$ *with* $\boldsymbol{\epsilon}_i \in \mathbb{R}^w$, $\boldsymbol{\epsilon}_1 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}(\lambda, w))$ *and*

$$\boldsymbol{\epsilon}_{i+1} = \sqrt{c}\boldsymbol{\epsilon}_i + \sqrt{1-c}\tilde{\boldsymbol{\epsilon}} \quad \textit{with} \quad \tilde{\boldsymbol{\epsilon}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}(\lambda, w)),$$

where $\boldsymbol{\Sigma}(\lambda, w)$ *is defined in equation 1. Then the combined* $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}(\lambda, c, w, n))$ *and*

$$\boldsymbol{\Sigma}(\lambda, c, w, n) = \begin{pmatrix} \boldsymbol{\Sigma}(\lambda, w) & \sqrt{c}\boldsymbol{\Sigma}(\lambda, w) & \cdots & (\sqrt{c})^{n-1}\boldsymbol{\Sigma}(\lambda, w)) \\ \sqrt{c}\boldsymbol{\Sigma}(\lambda, w) & \boldsymbol{\Sigma}(\lambda, w) & \cdots & (\sqrt{c})^{n-2}\boldsymbol{\Sigma}(\lambda, w) \\ \vdots & \vdots & \ddots & \vdots \\ (\sqrt{c})^{n-1}\boldsymbol{\Sigma}(\lambda, w) & (\sqrt{c})^{n-2}\boldsymbol{\Sigma}(\lambda, w) & \cdots & \boldsymbol{\Sigma}(\lambda, w) \end{pmatrix}.$$

Proof. For each $\boldsymbol{\epsilon}_{i+1}$ in the vector AR(1) model, it can be equivalently written as

$$\begin{aligned} \boldsymbol{\epsilon}_{i+1} &= \sqrt{c}\boldsymbol{\epsilon}_i + \sqrt{1-c}\tilde{\boldsymbol{\epsilon}}_{i+1} \\ &= \sqrt{c}(\sqrt{c}\boldsymbol{\epsilon}_{i-1} + \sqrt{1-c}\tilde{\boldsymbol{\epsilon}}_i) + \sqrt{1-c}\tilde{\boldsymbol{\epsilon}}_{i+1} \\ &= (\sqrt{c})^2\boldsymbol{\epsilon}_{i-1} + \sqrt{c(1-c)}\tilde{\boldsymbol{\epsilon}}_i + \sqrt{1-c}\tilde{\boldsymbol{\epsilon}}_{i+1} \\ &= (\sqrt{c})^2\boldsymbol{\epsilon}_{i-1} + \sqrt{1-c^2}\tilde{\boldsymbol{\epsilon}} \\ &= \cdots = (\sqrt{c})^i\boldsymbol{\epsilon}_1 + \sqrt{1-c^i}\tilde{\boldsymbol{\epsilon}}, \qquad \tilde{\boldsymbol{\epsilon}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}(\lambda, w)), \end{aligned}$$

where the fourth equation follows from the conclusion that the sum of two independent normal distributions is still a normal distribution. Then by similar arguments, we can get

$$\boldsymbol{\epsilon}_{i+1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}(\lambda, w))$$

Then for the combined noise $\boldsymbol{\epsilon}$, we have $\boldsymbol{\mu} = \mathbb{E}(\boldsymbol{\epsilon}) = \mathbb{E}(\boldsymbol{\epsilon}_1^\top, \boldsymbol{\epsilon}_2^\top, \cdots, \boldsymbol{\epsilon}_n^\top)^\top = \mathbf{0}$. And the covariance matrix is

$$\begin{aligned} \boldsymbol{\Sigma}(\lambda, c, w, n) &= \mathbb{E}[(\boldsymbol{\epsilon} - \boldsymbol{\mu})(\boldsymbol{\epsilon} - \boldsymbol{\mu})^\top] = \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top] = \mathbb{E}\left[\begin{pmatrix} \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \\ \vdots \\ \boldsymbol{\epsilon}_n \end{pmatrix} (\boldsymbol{\epsilon}_1 \quad \boldsymbol{\epsilon}_2 \quad \cdots \quad \boldsymbol{\epsilon}_n)\right] \\ &= \begin{pmatrix} \boldsymbol{\Gamma}(0) & \boldsymbol{\Gamma}(-1) & \cdots & \boldsymbol{\Gamma}(1-n) \\ \boldsymbol{\Gamma}(1) & \boldsymbol{\Gamma}(0) & \cdots & \boldsymbol{\Gamma}(2-n) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Gamma}(n-1) & \boldsymbol{\Gamma}(n-2) & \cdots & \boldsymbol{\Gamma}(0) \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{\Sigma}(\lambda, w) & \sqrt{c}\boldsymbol{\Sigma}(\lambda, w) & \cdots & (\sqrt{c})^{n-1}\boldsymbol{\Sigma}(\lambda, w)) \\ \sqrt{c}\boldsymbol{\Sigma}(\lambda, w) & \boldsymbol{\Sigma}(\lambda, w) & \cdots & (\sqrt{c})^{n-2}\boldsymbol{\Sigma}(\lambda, w) \\ \vdots & \vdots & \ddots & \vdots \\ (\sqrt{c})^{n-1}\boldsymbol{\Sigma}(\lambda, w) & (\sqrt{c})^{n-2}\boldsymbol{\Sigma}(\lambda, w) & \cdots & \boldsymbol{\Sigma}(\lambda, w). \end{pmatrix} \end{aligned}$$

where $\mathbb{E}(\boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_{i-k}^\top)$ is denoted as $\boldsymbol{\Gamma}(k)$ and $\boldsymbol{\Gamma}(-k)^\top = \boldsymbol{\Gamma}(k) = (\sqrt{c})^k\boldsymbol{\Gamma}(0)$, for positive k with $\boldsymbol{\Gamma}(0) = \boldsymbol{\Sigma}(\lambda, w)$. □

A.2 DDPM WITH CORRELATED NOISE

Let $\boldsymbol{\Sigma}$ be a valid covariance matrix. Then the forward and reverse diffusion in DDPM can be formulated as follows.

Forward diffusion: Given a data distribution $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$, $\boldsymbol{x}_t$ follows a Markov process with

$$
\begin{aligned}
q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) &= \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\boldsymbol{\Sigma}) \\
q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) &= \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \\
\boldsymbol{x}_t|\boldsymbol{x}_{t-1} \sim \mathcal{N}(\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\boldsymbol{\Sigma}) \;\; &\Leftrightarrow \;\; \boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \beta_t\boldsymbol{\epsilon}_{t-1}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})
\end{aligned}
$$

Let $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$, then

$$
\begin{aligned}
\boldsymbol{x}_t &= \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\boldsymbol{\epsilon} \; (\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})) \\
&= \cdots = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}.
\end{aligned}
$$

Then it gives

$$
\boldsymbol{x}_t|\boldsymbol{x}_0 \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, 1-\bar{\alpha}_t\boldsymbol{\Sigma})
$$

Reverse diffusion: By Bayes rule and Markov chain property, we have

$$
\begin{aligned}
q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) &= q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0)\frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \\
&\propto \exp\Bigg(-\frac{1}{2}\Bigg(\frac{(\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{x}_{t-1})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{x}_{t-1})}{\beta_t} + \frac{(\boldsymbol{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0)^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0)}{1-\bar{\alpha}_{t-1}} \\
&\quad - \frac{(\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0)^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0)}{1-\bar{\alpha}_t}\Bigg)\Bigg) \\
&= \exp\Bigg(-\frac{1}{2}\Bigg(\frac{\boldsymbol{x}_t^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{x}_t^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_{t-1} - \sqrt{\alpha_t}\boldsymbol{x}_{t-1}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_t + \alpha_t\boldsymbol{x}_{t-1}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_{t-1}}{\beta_t} \\
&\quad + \frac{\boldsymbol{x}_{t-1}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_{t-1}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_0 + \bar{\alpha}_{t-1}\boldsymbol{x}_0^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_0}{1-\bar{\alpha}_{t-1}} \\
&\quad - \frac{(\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0)^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0)}{1-\bar{\alpha}_t}\Bigg)\Bigg) \\
&= \exp\Bigg(-\frac{1}{2}\Bigg(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)\boldsymbol{x}_{t-1}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_{t-1} \\
&\quad - \left(\frac{\sqrt{\alpha_t}}{\beta_t}\left(\boldsymbol{x}_t^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_{t-1} + \boldsymbol{x}_{t-1}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_t\right) + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\left(\boldsymbol{x}_0^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_{t-1} + \boldsymbol{x}_{t-1}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_0\right)\right) + C(\boldsymbol{x}_0, \boldsymbol{x}_t)\Bigg)\Bigg)
\end{aligned}
$$

Thus,

$$
\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0 \sim \mathcal{N}\left(\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t, \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\boldsymbol{\Sigma}\right).
$$

Following Appendix B in (Sohl-Dickstein et al., 2015), we have

$$
L_{\text{VLB}} = \mathbb{E}_q[\underbrace{D_{\text{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_T))}_{L_T} + \sum_{t=2}^{T}\underbrace{D_{\text{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))}_{L_{t-1}}\underbrace{-\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)}_{L_0}],
$$

and

$$
\begin{aligned}
L_{\text{VLB}} &= L_T + L_{T-1} + \cdots + L_0 \\
\text{where } L_T &= D_{\text{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_T)) \\
L_t &= D_{\text{KL}}(q(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}, \boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})) \text{ for } 1 \leq t \leq T-1 \\
L_0 &= -\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)
\end{aligned}
$$

In L_t, the two compared distributions are still Gaussian. Apply the simplifications in Ho et al. (2020), the loss function L_{simple} still holds.

B Experiment Details

B.1 Efficient method to construct covariance matrix

We present below the python code for constructing the covariance matrix.

```
def construct_cov_mat(n_frames,decay_rate,theta=0):
    seq = torch.pow(decay_rate,torch.arange(num_frames))
    if theta != 0:
        seq = torch.multiply(seq,torch.cos(torch.arange(n_frames)*theta))
    return toeplitz(seq,seq)

def toeplitz(c, r):
    vals = torch.cat((r, c[1:].flip(0)))
    shape = len(c), len(r)
    i, j = torch.ones(*shape).nonzero().T
    return vals[j-i].reshape(*shape)
```

B.2 Selection of Hyper-parameters

Table 5: Selected hyper-parameters of decay rate λ, coefficient c in the AR(1) model, and window size w on both audio and motion generation tasks: neural vocoding and unconditional generation tasks for audio, and unconditional generation, action-to-motion generation and text-to-motion generation for motion.

Task/Problem	Model	λ	c	w
Audio generation				
Neural vocoding	DiffWave + ARTDiff	0.1	0.1	16
	DiffWave-sashimi + ARTDiff	0.1	0.1	16
Unconditional generation	DiffWave + ARTDiff	0.1	0.1	16
	DiffWave-sashimi + ARTDiff	0.1	0.1	16
Motion generation				
Unconditional generation	MDM + ARTDiff	0.05	-	-
Action-to-motion generation (Humanact12)	MDM + ARTDiff	0.1	-	-
Text-to-motion generation (HumanML3D)	MDM + ARTDiff	0.1	0.1	49
Text-to-motion generation (KIT)	MDM + ARTDiff	0.1	0.1	49

In the one-shot text-to-video generation task, we propose to grid search over the combinations of $\lambda(\gamma) \in \{0.5, 0.6, 0.7, 0.8\}, \eta \in \{0.4, 0.6, 0.8\}, \theta \in \{0, 0.05, 0.1, 0.15, 0.2\}$. And the coefficient for controlling the random noise added in DDIM inversion is searched in $\tau \in \{0, 0.1, 0.2, 0.3, 0.4\}$.

B.3 Dataset and prompts in one-shot text-to-video generation

We select 8 reference videos from the example videos in Wu et al. (2023) and the DAVIS dataset (Pont-Tuset et al., 2017), covering a range of categories including animals, vehicles, and humans. The selected video names are *blackswan, car-turn, hike, kite-surf, mallard-water, man-skiing, man-surfing, rabbit-watermelon*. To obtain video caption, we use BLIP-2 (Li et al., 2023) for automated video captioning. We then manually design four edited prompts for each video, resulting 32 edited prompts and 32 generated videos in total. These edited prompts include object editing, background changes, and style transfers.