STRUCTURE-AWARE COMPRESSION DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

The conventional reverse diffusion process focuses on the log-likelihoods based denoising, which may neglect the intrinsic laws in the natural world. In this work, we propose the Structure-aware Compression Diffusion Models (SCDM), with a novel reverse diffusion process that enforces the lifting and compression operation towards learning the low-dimensional structures of the original data. Our SCDM outperforms the conventional diffusion models on both physical datasets and image datasets. Besides, the experiments demonstrate that the low-dimensional structures learned by SCDM align well with scientific theories in several physical data generation tasks.

1 INTRODUCTION

Diffusion models have emerged as a powerful class of generative models to generate synthetic samples from the learned data distributions, advancing the data synthesis in the field of computer vision Ho et al. (2020); Song et al. (2020a;b); Dhariwal & Nichol (2021). These successes suggest the great potential to build a data recovery model along with the denoising process. However, a conventional reverse diffusion process merely overfits a data denoising objective. In the scenarios, the conventional denoising objective function does not explicitly encourage the model to preserve the underlying low-dimensional structures. As shown in Figure 1, we compare the generation results of a representative diffusion model, DDPM (Ho et al., 2020), and our Structure-aware Compression Diffusion Model (SCDM) on two structured datasets: the circle dataset in (x, y) space and the motion-tracked pendulum dataset in (θ, ω) space.

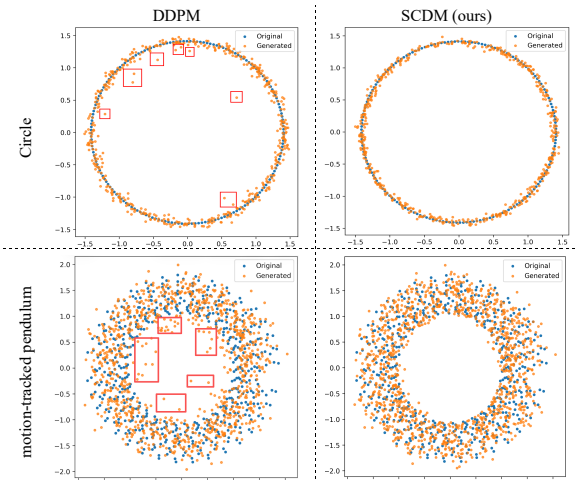


Figure 1: Comparison of the DDPM and our SCDM on two toy generation tasks with clear physical meanings. The red boxes mark the improperly generated samples. Given the same training data (blue dots) and the same learning epochs, the generated samples (orange dots) visualize the advantage of our SCDM over the conventional DDPM.

The underlying physical processes are governed by a small number of low-dimensional structures and interactions Schmidt & Lipson (2009); Brunton et al. (2016). For example, we could observe plenty of data from a swinging pendulum experiment, but all observations should be governed by a second-order ordinary differential equation with finite variables. For scientists to find these concise yet powerful analytical low-dimensional structures, theories are derived from observations, which subsequently guide the induction of hypothesis and further inferences. In this work, we explore the possibility to train a generative model that is implicitly restricted by the learned low-dimensional structures, in imitation of the scientists' induction and inference process. The learned generative model could be further used to generate robust and reliable synthetic data.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

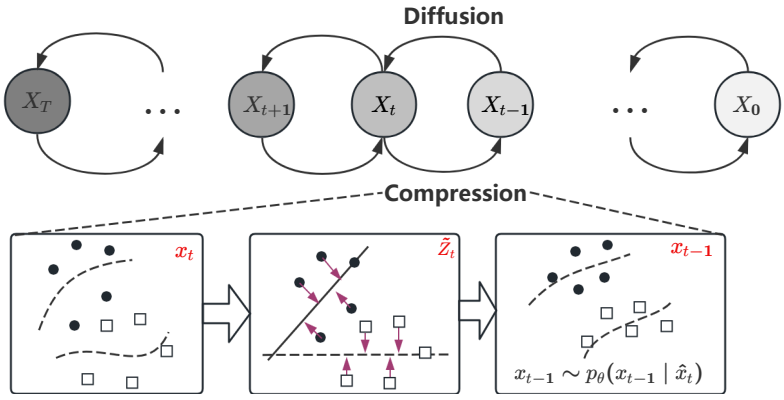


Figure 2: Overview of the process in the Structure-aware Compression Diffusion Models (SCDM). The forward arrow illustrates the conventional diffusion process, where x_0 is gradually corrupted into x_T by successive noise injection. In contrast to the standard reverse process, the backward arrow illustrates a compression-guided reverse process, where structured representations $\tilde{Z}_t$ are constructed to guide denoising from x_t to x_{t-1}.

We propose the Structure-aware Compression Diffusion Models (SCDM), which integrates diffusion models with representation learning. In contrast to the conventional diffusion models, the SCDM introduces a compression objective during the reverse diffusion process. The intuition is that the diffused data should be denoised under some low-dimensional structures, so that the recovered data could be robust. Our SCDM leverages the conventional diffusion model's capabilities of data recovery, while simultaneously aligning with scientific hypotheses of the low-dimensional structures.

Figure 2 illustrates the SCDM, which consists of two processes: a conventional forward diffusion process that progressively adds noise to original data $x_0 \to x_T$, and a novel compression process that iteratively removes noise by uncovering the low-dimensional structures underlying the data. Specifically, during the compression process, the noisy representation x_t is projected onto a learnable low-dimensional subspace to obtain a structure-aware representation $\tilde{Z}_t$. Representation $\tilde{Z}_t$ is then fed into the denoising network to predict the noise and guide the reverse diffusion step.

The contributions of this paper are two-fold:

- We introduce Structure-aware Compression Diffusion Models (SCDM), a novel framework that integrates structure-aware representation into diffusion modeling, restricting the reverse process to learn the low-dimensional structures.
- Empirical results on physical data synthesis demonstrate that the low-dimensional structures learned by SCDM are well aligned with scientific theories. Besides, SCDM also exhibits robustness when the training data is partially corrupted.

2 Related Work

2.1 Structural Priors for Scientific Data

In scientific domains, observations are often governed by a limited number of underlying mechanisms Holmes et al. (1996); Schmid (2010); Kutz et al. (2016). Human motion data can often be summarized by a small set of coordinated movement patterns, while physical dynamical systems are frequently controlled by a few essential state variables Lee & Elgammal (2006); Brunton et al. (2016). Similarly, robotics, sensing, and biological measurements often evolve on low-dimensional manifolds induced by system constraints or latent organizational principles Rhodes (2024); Carvalho et al. (2007); Safonova et al. (2004). To recover these underlying structures from observations, classical methods such as dictionary learning, sparse approximation, and subspace-based modeling Olshausen & Field (1996); Aharon et al. (2006); Mairal et al. (2009); Wright et al. (2009); Elhamifar & Vidal (2013) exploit such structural regularities by representing high-dimensional observations with a limited set of basis elements, sparse coefficients, or low-dimensional subspaces.

2.2 Guided and Constrained Diffusion Models

Classifier guidance improves conditional sample fidelity by injecting classifier gradients into the reverse process, while classifier-free guidance achieves a similar effect without training a separate classifier Ho & Salimans (2022); Dhariwal & Nichol (2021); Han et al. (2022). Beyond semantic guidance, ILVR Choi et al. (2021) refines the denoising trajectory using low-frequency information from a reference image, enabling training-free conditional generation. In scientific machine learning, constrained diffusion methods incorporate physics-based priors or observational constraints to improve physical consistency, including PDE-guided diffusion, Physics-Informed Diffusion Models (PIDM), and diffusion models for PDE simulations under partial observation Shu et al. (2023); Bastek et al. (2024); Shysheya et al. (2024).

2.3 Structured Representation Learning for Diffusion Models

Learning compact representations has become an important strategy for improving the efficiency and effectiveness of diffusion models Kingma & Welling (2014); Diederik & Max (2019); Karras et al. (2019); Brock et al. (2018); Zheng et al. (2025); Kim et al. (2024). A common direction is to perform diffusion in a learned latent space. For example, LSGM combines variational autoencoders with score-based generative modeling, while LDM first compresses images into a perceptual latent space and then performs diffusion in this lower-dimensional representation space Vahdat et al. (2021); Rombach et al. (2022). Another direction treats diffusion models as representation learners. Diffusion Autoencoders learn semantic representations with an encoder and use a diffusion model to decode stochastic details, while SODA introduces a bottleneck representation to guide denoising-based view generation Preechakul et al. (2022); Hudson et al. (2024). Different from these methods, we integrate representation learning into the denoising process itself, allowing the model to progressively learn the intrinsic compact structure of the input data.

3 Preliminary

3.1 Denoising Diffusion Probabilistic Models

Denoising Diffusion Probabilistic Models (DDPM) Ho et al. (2020) are a class of generative models, which consists of a forward and a reverse diffusion process. In the forward process, Gaussian noise is gradually added to the clean data x_0 over T timesteps, forming a Markov chain:

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1}), \tag{1}$$

where $q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}\, x_{t-1}, \beta_t \mathbf{I})$, α_t and β_t denote the variance-preserving noise schedule parameters. The reverse process aims to denoise x_t back to x_0 by learning a noise prediction network $\epsilon_\theta(x_t, t)$. The training objective is to minimize the discrepancy between the true noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the predicted noise, formulated as:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}_0,\, \boldsymbol{\epsilon},\, t} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(\sqrt{\bar{\alpha}_t}\, \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\, \boldsymbol{\epsilon},\, t \right) \right\|^2 \right], \tag{2}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ is the cumulative product of noise schedules. New samples are generated by starting from pure Gaussian noise and iteratively applying the reverse denoising process.

3.2 Compression via sparse rate reduction

Motivated by the idea that effective representations should capture essential structure while discarding redundancy Brunton et al. (2016), sparse rate reduction (CRATE) Yu et al. (2023) is primarily designed to learn discriminative representations from data containing multiple semantic structures. It assumes that token representations $z_i \in Z$ lie in a mixture of low-dimensional Gaussian subspaces with bases $\{U_j\}_{j=1}^{N}$, and optimizes the representation function $Z = f(X)$ by:

$$\max_{f \in \mathcal{F}} \mathbb{E}_Z \left[R(Z) - R^c(Z; U_{[N]}) - \lambda \|Z\|_0 \right], \tag{3}$$

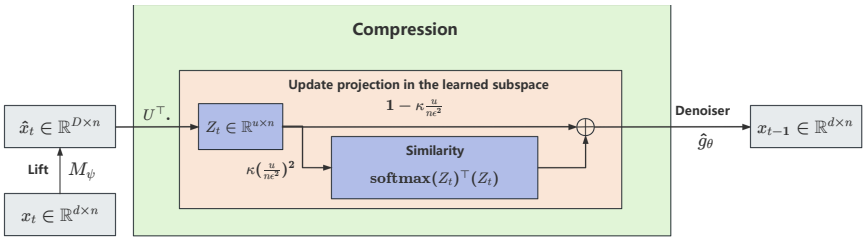


Figure 3: Overview of the structure-aware compression module in SCDM.

where $R(Z)$ denotes the total coding cost of Z, $R^c(Z;U_{[N]})$ is the coding rate under a mixture of N low-dimensional Gaussian subspaces, and $\|Z\|_0$ encourages sparsity. This coding-rate reduction promotes compact and discriminative representations across different subspaces and is fully mathematically interpretable. Inspired by this principle, we adopt rate-based compression as a structural prior. Unlike CRATE, however, our objective is to capture the common underlying dynamical laws shared by samples, and thus we employ a single shared low-dimensional subspace.

4 METHODOLOGY

4.1 MODEL FRAMEWORK

From the forward diffusion process, the noisy sample at timestep t is given by

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon, \qquad \varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{4}$$

Hence, taking the conditional expectation with respect to x_t, we have

$$\mathbb{E}\big[\varepsilon \,\big|\, x_t\big] = \mathbb{E}\Big[\frac{x_t - \sqrt{\bar{\alpha}_t}\, x_0}{\sqrt{1-\bar{\alpha}_t}} \,\Big|\, x_t\Big] = \frac{1}{\sqrt{1-\bar{\alpha}_t}}\Big(x_t - \sqrt{\bar{\alpha}_t}\, \mathbb{E}[x_0 \mid x_t]\Big). \tag{5}$$

The quantity of interest is precisely the structural information in x_t that determines the posterior mean $\mathbb{E}[x_0 \mid x_t]$. Consequently, as long as a representation extracted from x_t preserves the low-dimensional structures of x_0 that govern $\mathbb{E}[x_0 \mid x_t]$, it is sufficient for accurate ε-prediction as well.

Motivated by the above observation, we seek a representation transformation that extracts from x_t the low-dimensional structures of x_0 relevant to predicting ε (as illustrated in Fig. 3). However, the intrinsic dimensionality and form of an appropriate denoising representation are generally unknown a prior. Likewise, it is unclear which features or feature combinations are most informative for noise prediction. We therefore use lifting to map the observations into a redundant feature space containing richer nonlinear combinations of the original variables. As the number of the lifted dimensions increases, there is higher probability to come up with more helpful combined features that is aligned with the nature of the tasks.

Specifically, let $x_t \in \mathbb{R}^{d\times n}$ denote the noisy representation at diffusion timestep t, where d denotes the feature dimension of each sample and n is the number of samples. Inspired by prior works that learn high-dimensional embeddings for structured modeling Qian et al. (2020); Lusch et al. (2018); Takeishi et al. (2017), we use a simple neural network to lift the original noisy state into a higher-dimensional feature space:

$$\hat{x}_t = M_\psi(x_t), \quad M_\psi : \mathbb{R}^d \to \mathbb{R}^D, \tag{6}$$

where D is the lifted feature dimension and ψ denotes the learnable parameters of the lifting network. This lifting operation provides a redundant feature space in which the intrinsic low-dimensional structure of the noisy representation can be more easily identified. We typically choose a relatively large lifted dimension D to avoid a high-variance latent space. The robustness of this setting is further verified in the Section 5.2.1.

To capture low-dimensional structure from $\hat{x}_t$, we introduce a learnable subspace basis matrix $U \in \mathbb{R}^{D\times u}$, where u is the subspace dimension. The projected representation is given by $Z_t = U^\top \hat{x}_t \in \mathbb{R}^{u\times n}$. The subspace U is jointly optimized with the denoising model parameters. Although the

Algorithm 1 Training and Sampling Procedure of SCDM

Training Phase
Sample x_0, $t \sim \mathrm{Uniform}\{1, \dots, T\}$, and $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$x_t \leftarrow \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon$ ▷ Forward diffusion
$\hat{x}_t \leftarrow M_\psi(x_t)$ ▷ Lift x_t into the higher-dimensional feature space
$Z_t \leftarrow U^\top \hat{x}_t$ ▷ Project onto the learned subspace
$\tilde{Z}_t \leftarrow Z_t \left[\left(1 - \kappa \frac{u}{n\epsilon^2}\right) I + \kappa \left(\frac{u}{n\epsilon^2}\right)^2 \mathrm{softmax}\big(Z_t^\top Z_t\big) \right]$ ▷ compression using equation 10
$\hat{\varepsilon} \leftarrow \hat{g}_\theta(\tilde{Z}_t, t)$ ▷ Noise prediction
$(\psi, \theta, U) \leftarrow (\psi, \theta, U) - \eta \nabla_{\psi,\theta,U} \|\varepsilon - \hat{\varepsilon}\|_2^2$ ▷ Joint optimization using equation 12
Sampling Phase
$x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Initialize from Gaussian prior
for $t = T$ **to** 1 **do**
  $\hat{x}_t \leftarrow M_\psi(x_t)$ ▷ Lift x_t into the higher-dimensional feature space
  $Z_t \leftarrow U^\top \hat{x}_t$ ▷ Project onto the learned subspace
  $\tilde{Z}_t \leftarrow Z_t \left[\left(1 - \kappa \frac{u}{n\epsilon^2}\right) I + \kappa \left(\frac{u}{n\epsilon^2}\right)^2 \mathrm{softmax}\big(Z_t^\top Z_t\big) \right]$ ▷ compression using equation 10
  $\hat{\varepsilon} \leftarrow \hat{g}_\theta(\tilde{Z}_t, t)$ ▷ Predict current noise
  $\mu_\theta \leftarrow \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\, \hat{\varepsilon} \right)$ ▷ Reverse-process mean
  $\gamma \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ **if** $t > 1$ **else** $\gamma \leftarrow 0$ ▷ Reverse-step stochasticity
  $x_{t-1} \leftarrow \mu_\theta + \sigma_t \gamma$ ▷ One-step reverse update using equation 11
end for
return x_0

learned subspace U captures denoising-relevant directions, the projected representation Z_t may still contain redundant and highly correlated variations. A desirable representation should be compact and non-redundant. We use the lossy coding rate Chan et al. (2022) to measure how many degrees of freedom are required to encode the projected noisy representation Z_t up to precision $\epsilon > 0$:

$$R(Z_t) = \frac{1}{2} \log\det\Big(I + \frac{u}{n\epsilon^2} Z_t^\top Z_t\Big). \tag{7}$$

A lower coding rate indicates that the representation can be described more compactly within the subspace. Accordingly, we compress Z_t by moving it along the negative gradient direction of the coding-rate objective:

$$\tilde{Z}_t = Z_t - \kappa \nabla_{Z_t} R(Z_t) = Z_t - \kappa \frac{u}{n\epsilon^2} Z_t \left(I + \frac{u}{n\epsilon^2} Z_t^\top Z_t\right)^{-1}, \tag{8}$$

where κ is learning rate. Computing the inverse $\left(I + \frac{u}{n\epsilon^2} Z_t^\top Z_t\right)^{-1}$ is expensive Chan et al. (2022). Following CRATE Yu et al. (2023), we approximate $\nabla_{Z_t} R(Z_t)$ with the first order term of its von Neumann expansion as:

$$\nabla_{Z_t} R(Z_t) \;\approx\; \frac{u}{n\epsilon^2} Z_t \;-\; \left(\frac{u}{n\epsilon^2}\right)^2 Z_t\, \mathrm{softmax}\big(Z_t^\top Z_t\big). \tag{9}$$

The $\mathrm{softmax}\big(Z_t^\top Z_t\big)$ introduces similarity-aware interactions among projected samples through normalized pairwise similarity weights, allowing samples to share information in the learned subspace.

Therefore, we can rewrite equation 8 as

$$\begin{aligned} \tilde{Z}_t &\approx Z_t - \kappa \frac{u}{n\epsilon^2} Z_t + \kappa \left(\frac{u}{n\epsilon^2}\right)^2 Z_t\, \mathrm{softmax}\big((Z_t)^\top (Z_t)\big) \\ &= Z_t \left[\left(1 - \kappa \frac{u}{n\epsilon^2}\right) I + \kappa \left(\frac{u}{n\epsilon^2}\right)^2 \mathrm{softmax}\big(Z_t^\top Z_t\big) \right]. \end{aligned} \tag{10}$$

Finally, we feed the refined representation $\tilde{Z}_t$ into the denoising network $\hat{g}_\theta(\tilde{Z}_t, t)$. Then the reverse diffusion step follows the standard DDPM parameterization:

$$x_{t-1} \sim p_\theta(x_{t-1} \mid x_t, \tilde{Z}_t) = \mathcal{N}\Big(\mu_\theta(\tilde{Z}_t, t), \sigma_t^2 \mathbf{I}\Big), \tag{11}$$

Table 1: Sampling performance across seven motion datasets. CT: unit circle; DL: double linear oscillator; LA: single linear oscillator; PA: pendulum angular; PC: pendulum Cartesian; LH: single linear oscillator without acceleration; PH: pendulum angular without acceleration.

Metric	Model	CT	DL	LA	PA	PC	LH	PH
RMAE ($\times 10^{-2}$)	SCDM	**1.37**	19.81	**24.59**	**33.76**	49.44	**10.17**	**27.41**
	DDPM	4.00	19.99	26.67	35.21	47.38	10.29	28.05
	DDIM	4.01	20.09	33.02	49.12	62.19	11.06	31.41
	SDD	2.67	**19.64**	26.06	72.85	**43.78**	10.44	27.65
	PIDM	2.27	54.69	52.23	77.40	64.16	-	-
MDN %CE	SCDM	**4.14**	**6.78**	**7.54**	**1.95**	**3.49**	**0.94**	**2.67**
	DDPM	7.21	7.09	8.61	3.47	6.97	7.57	3.64
	DDIM	18.15	38.58	33.76	17.95	40.87	31.16	37.86
	SDD	4.76	12.57	14.63	2.49	15.08	5.84	4.44
	PIDM	11.13	81.26	10.46	13.95	3.71	-	-
%VFE	SCDM	**+0.05**	**-0.02**	+1.64	+4.33	**+1.51**	+1.21	+1.29
	DDPM	+0.39	+0.07	**-1.09**	**+0.41**	+8.38	**+0.49**	**-0.65**
	DDIM	+0.34	+1.08	+9.98	+28.36	+54.77	+6.23	+22.12
	SDD	-0.05	-1.14	-1.25	-1.85	+8.44	+1.59	+1.68
	PIDM	+2.74	-5.66	-6.17	-17.17	-36.24	-	-

where $\mu_\theta(\tilde{Z}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\hat{g}_\theta(\tilde{Z}_t, t)\right)$. During training, the parameters of the lifting network ψ, the subspace U, and the denoising network θ are jointly optimized under the standard noise prediction objective:

$$\mathcal{L} = \min_{\theta,\psi,U} \mathbb{E}_{x_0,\varepsilon,t}\left[\left\|\varepsilon - \epsilon_\theta\left(C_\kappa\left(U^\top \phi_\psi(x_t)\right), t\right)\right\|_2^2\right], \tag{12}$$

where C_κ denotes the coding-rate compression operator refer to equation 10.

5 Experiment

5.1 Performance of Our SCDM

Physical Datasets The datasets are from the Supporting Online Material (SOM) [1] of Schmidt & Lipson (2009). These data were recorded on standard tabletop mechanics setups (air track oscillators and single and double pendulum rigs) using video tracking.

To be consistent, we follow the conventional symbol noting system in Physics. We test on six scenarios: (1) double linear oscillator (DL), recorded (x_1, x_2, v_1, v_2) on 495 points; (2) single linear oscillator (LA), recorded (x, v, a) on 870 points; (3) pendulum in angular coordinates (PA), recorded (θ, ω, α) on 556 points; (4) pendulum in Cartesian coordinates (PC), recorded (x, y) on 600 points. To further examine the performance of the model when explicit physical information is incomplete, we additionally introduce two reduced datasets without acceleration: (5) single linear oscillator without acceleration (LH), recorded (x, v) on 870 points; (6) pendulum in angular coordinates without acceleration (PH), recorded (θ, ω) on 556 points.

The data are normalized to zero mean and unit variance before training. For SCDM, the noisy input x_t is first lifted to a 64-dimensional latent representation X_t through a simple learnable lifting network. The diffusion process adopts the cosine β-schedule ($T = 1000$, $\beta_1 \approx 10^{-5}$ to $\beta_T \approx 0.02$). The Fourier embedding of timestep t is concatenated with $\widetilde{X}_t$ and passed through a three-layer GELU-MLP decoder to predict the noise. AdamW was adopted in training ($lr = 1e-3$, $wd = 1e-4$). EMA (decay = 0.995) was introduced to enhance the stability.

Baselines. We compared our SCDM with four baselines. **DDPM** Ho et al. (2020) is a classical probabilistic diffusion model, which adopts a forward Markov noising process and learn a reverse denoising model for generation. **DDIM** Song et al. (2020a) is a non-Markovian deterministic variant of DDPM that accelerates sampling. **SDD** Ostheimer et al. (2025) augments each dimension with a binary sparsity bit that is diffused alongside the data. **PIDM** Bastek et al. (2024) augments diffusion

[1] http://www.sciencemag.org/cgi/content/full/324/5923/81/DC1

Table 2: Stability of generation under different lifted dimensions D on the single linear oscillator (LA) dataset. The subspace dimension is fixed to $u = 6$. Results are reported as mean $\pm$ std over thirty random seeds.

D	RMAE ($\times 10^{-2}$)	MDN %CE	%VFE
8	1115.23 ± 2995.80	1063.86 ± 2873.34	$+910747.44 \pm 2738803.69$
12	1353.74 ± 7275.46	1335.20 ± 7274.35	$+76547446.03 \pm 419267640.61$
16	25.29 ± 0.70	6.12 ± 3.21	-1.02 ± 2.18
32	25.27 ± 0.71	6.36 ± 3.44	$+0.64 \pm 1.73$
64	25.22 ± 0.69	6.72 ± 3.42	$+1.36 \pm 1.17$
128	25.15 ± 0.64	6.13 ± 3.45	$+1.35 \pm 1.09$

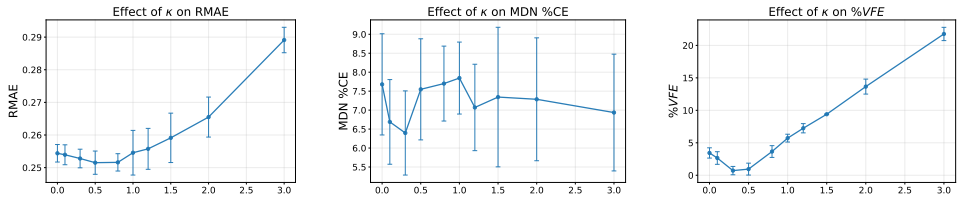


Figure 4: Effect of the compression step size κ on generation quality. The blue curve denotes the mean across random seeds, and the error bars indicate one standard deviation.

with physics-informed residuals to enforce governing equations, providing a complementary form of inductive bias.

To verify the correctness of the generated data, we use three established metrics Bastek et al. (2024): average radius absolute error (R_{MAE}), center offset percentage (MDN%CE), and area error percentage (% VFE). All methods share the same MLP-based denoising backbone for fair comparison. The results are summarized in Table 1. SCDM demonstrates strong overall performance and obtains the best results on a majority of datasets across multiple evaluation metrics. On the PC dataset, the decaying pendulum motion leads to a strongly nonuniform sample distribution, with substantially fewer observations near the outer endpoints (see Appendix Figure 7), which partly accounts for the observed RMAE difference. Nevertheless, SCDM achieves the best MDN %CE and %VFE on PC, demonstrating a favorable overall trade-off across the three evaluation metrics.

5.2 Analytical Structure Awareness by our SCDM

Our SCDM achieves structure awareness by lifting the original dimension of the observed data, then learning a compressed structure. We analyze the stability of the lifted dimension and the learned subspaces as follows. In addition, a more detailed ablation study of the key components of SCDM is provided in Appendix E.

5.2.1 Stability of the Lifted Dimension

As shown in Table 2, increasing the number of the lifted dimensions generally improves the generation performance. This observation is consistent with the purpose of the lifting operation, i.e. the lifting operation projects the samples to a redundant feature space, which provides more combinations of the original features. As the number of the lifted dimensions increases, there is higher probability to come up with more helpful combined features that is aligned with the nature of the tasks. The experimental results are presented in the Appendix K. We tested each experimental setting with 30 random seeds and verified our assumption.

5.2.2 Sensitivity to Compression Step Size and Subspace Dimension

As shown in equation 10, the compression behavior is mainly controlled by the step size κ and the subspace dimension u. κ determines the update strength, while u affects both the scaling factor $\frac{u}{n\epsilon^2}$

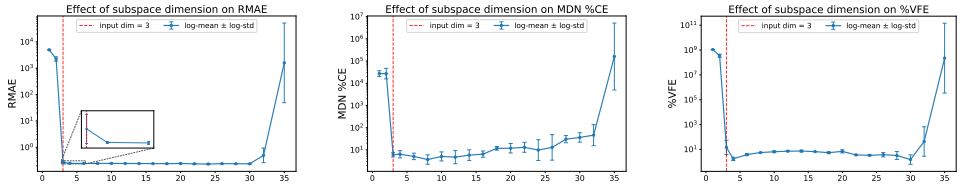
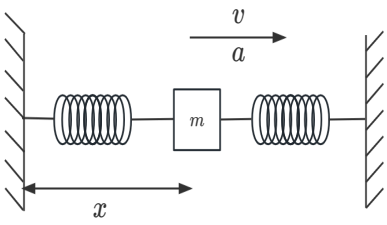


Figure 5: Effect of the subspace dimension u on generation quality. The blue curve denotes the geometric mean across random seeds, and the error bars indicate one standard deviation in logarithmic space. The red dashed line marks the input state dimension.

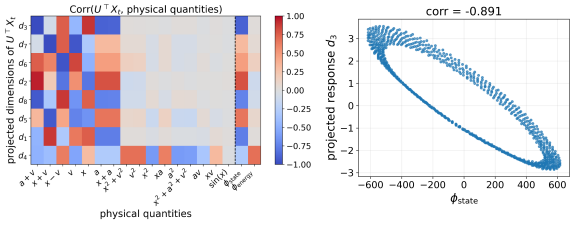


System: single linear oscillator

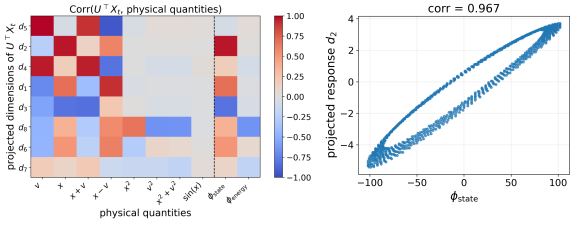
ϕ_{state}: $a = 6.03x - 0.01v - 0.03$

ϕ_{energy}: $x^2 + v^2$

Even without explicit acceleration a in the observations, subspace dimensions can still recover the state quantity from x and v.



（a）single linear oscillator with acceleration (x,v,a)



（b）single linear oscillator without acceleration (x,v)

Figure 6: Interpretability analysis of dimensions in learned low-dimensional subspace for physical system: the single linear oscillator (LA). In each panel, the left part shows the system schematic, the middle part reports the correlation heatmap between dimensions in learned subspace and candidate physical targets, including state equation ϕ_{state} and energy equation ϕ_{energy}. The right part visualizes the representative relation between a selected dimension and state equation.

and the dimension of the projected representation $Z_t = U^\top \hat{x}_t \in \mathbb{R}^{u \times n}$. We use the single linear oscillator (LA) as the primary case, train each setting with five random seeds.

First, we investigate the influence of the compression learning rate κ. As shown in Fig. 4, moderate compression ($\kappa = 0.3$–0.5) provides the best overall trade-off across the three metrics. In contrast, excessive compression degrades generation quality, suggesting the loss of useful information.

Then, we investigate the influence of the subspace dimension u on LA. Figure 5 shows that moderate values of u achieve the best trade-off between representation capacity and structural constraint. A similar trend is observed on the other physical datasets, with results provided in Appendix H.

5.2.3 Subspace Structural Analysis

We further investigate the interpretability of dimensions in learned low-dimensional subspace under the linear oscillator (LA). For each dimension, we use its mean response as a scalar summary, so that it can be directly compared with scalar physical variables or formula-based targets. As an additional example, the corresponding interpretability analysis on PA is provided in Appendix I.

Since acceleration is determined by the underlying state equation, we compare settings with and without acceleration to test whether the learned representation can recover latent physical structure from the remaining state variables. As shown in Figure 6, even when acceleration a is not included in the observations, several dimensions in the learned subspace still show strong correlations with

Table 3: FID and IS on CIFAR-10 under different levels of noisy training data. The last column reports the FID degradation relative to the clean-training setting (0% noise) for each model.

Noise (%)	Model	FID ↓	IS ↑	FID Degradation ↓
0	SCDM	**2.86**	**9.95 ± 0.13**	-
0	DDPM	3.17	9.46 ± 0.11	-
0	SC-VAE	4.41	6.22 ± 0.08	-
0	Denoising Diffusion GAN	3.75	9.63 ± 0.11	-
30	SCDM	**22.75**	**9.17 ± 0.14**	**19.89**
30	DDPM	30.55	8.77 ± 0.08	27.38
30	SC-VAE	45.93	5.99 ± 0.07	41.52
30	Denoising Diffusion GAN	25.56	8.71 ± 0.14	21.81
40	SCDM	**32.96**	**8.49 ± 0.16**	**30.10**
40	DDPM	43.90	8.15 ± 0.13	40.73
40	SC-VAE	49.01	5.79 ± 0.10	44.60
40	Denoising Diffusion GAN	38.56	8.19 ± 0.12	34.81
50	SCDM	**51.16**	**7.81 ± 0.09**	**48.30**
50	DDPM	59.67	7.44 ± 0.11	56.50
50	SC-VAE	53.11	5.76 ± 0.08	48.70
50	Denoising Diffusion GAN	57.37	6.51 ± 0.09	53.62

raw variables and physically meaningful formula targets. This indicates that the learned subspace representation captures physical structure at a concept level, rather than merely reflecting individual variable responses.

5.3 Robustness of Our SCDM

Datasets To evaluate the robustness of SCDM under noisy training data and to investigate the generalization to image generation tasks, we conduct experiments on CIFAR-10. We construct multiple corrupted training sets by randomly selecting a certain proportion of training images (Noise = 0%, 30%, 40%, 50%) and adding zero-mean Gaussian noise with standard deviation σ in pixel space, followed by clipping to [0,1]. The Inception Score and FID Heusel et al. (2017) are computed with clean image set.

Baselines. DDPM Ho et al. (2020). **SC-VAE** Xiao et al. (2025) is a variational auto-encoder with LISTA-style sparse encoders. **Denoising Diffusion GAN** Xiao et al. (2021) combines the iterative denoising mechanism of diffusion models with adversarial training.

For image data, following CRATE Yu et al. (2023), we reshape the U-Net bottleneck features as $X_t \in \mathbb{R}^{D\times n}$, where D and n denote the channel and spatial dimensions, respectively. SCDM is then applied to this matrix, with subspace projection along the channel dimension and similarity interaction among spatial positions. For CIFAR-10, the $256 \times 4 \times 4$ U-Net bottleneck feature map is reshaped to $X_t \in \mathbb{R}^{256\times 16}$, lifted to $\phi(X_t) \in \mathbb{R}^{512\times 16}$, and projected by $U \in \mathbb{R}^{512\times 32}$ to $Z_t \in \mathbb{R}^{32\times 16}$. We use one compression update using $\epsilon = 3.0$ and $\kappa = 0.5$.

The results are summarized in Table 3. As the corruption level increases, the performance of all methods degrades; however, SCDM exhibits consistently smaller degradation and maintains a clearer advantage over the baselines under noisy training conditions. Visual examples of generated samples under different noise levels are provided in Appendix J.

6 Conclusion

In this paper, we propose the SCDM, a novel framework that integrates structured representation learning into diffusion models to learn the meaningful low-dimensional structures in the reverse diffusion process. The implicitly learned structure-aware representations improves the generation quality and better aligns with underlying intrinsic laws in some typical physics experiments. Empirical results on CIFAR-10 and several physical datasets demonstrate that the structure-aware compression mechanism is a promising direction to research on.

AI USE STATEMENT

Generative AI tools were used solely for language editing and improving the readability of the manuscript, without affecting the core methodology, scientific rigor, or originality of the research. This research does not involve LLMs as any core, original, or nonstandard component. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Detailed experimental settings and implementation details are provided within each experimental section, covering both the main experiments and ablation studies. In addition, executable code is provided in the supplementary materials to further facilitate reproduction of the reported results.

REFERENCES

M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis M Kochmann. Physics-informed diffusion models. *arXiv preprint arXiv:2403.14404*, 2024.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, March 2016.

Schubert R Carvalho, Ronan Boulic, and Daniel Thalmann. Interactive low-dimensional human motion synthesis by combining motion models and pik. *Computer Animation and Virtual Worlds*, 18(4-5):493–503, 2007.

Kwan Ho Ryan Chan, Yaodong Yu, Chong You, Haozhi Qi, John Wright, and Yi Ma. Redunet: a white-box deep network from the principle of maximizing rate reduction. *J. Mach. Learn. Res.*, 23(1), January 2022.

Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

P Kingma Diederik and Welling Max. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. Card: Classification and regression diffusion models. *Advances in Neural Information Processing Systems*, 35:18100–18115, 2022.

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 and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020.

Philip Holmes, John L. Lumley, and Gal Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge Monographs on Mechanics. Cambridge University Press, 1996.

Drew A Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K Lampinen, Andrew Jaegle, James L McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. Soda: Bottleneck diffusion models for representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23115–23127, 2024.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Yeongmin Kim, Kwanghyeon Lee, Minsang Park, Byeonghu Na, and Il-Chul Moon. Diffusion bridge autoencoders for unsupervised representation learning. *arXiv preprint arXiv:2405.17111*, 2024.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

Chan-Su Lee and Ahmed Elgammal. Human motion synthesis by motion manifold learning and motion primitive segmentation. In *International Conference on Articulated Motion and Deformable Objects*, pp. 464–473. Springer, 2006.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pp. 689–696, 2009.

Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, June 1996.

Phil Ostheimer, Mayank Nagda, Jean Radig, Carl Herrmann, Stephan Mandt, Marius Kloft, and Sophie Fellenz. Sparse data generation using diffusion models. *arXiv preprint arXiv:2502.02448*, 2025.

Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10619–10629, 2022.

Elizabeth Qian, Boris Kramer, Benjamin Peherstorfer, and Karen Willcox. Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. *Physica D: Nonlinear Phenomena*, 406:132401, 2020.

Travers Rhodes. *Learning Low-Dimensional Latent Representations of Demonstrated Trajectories for Robots*. Cornell University, 2024.

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

Alla Safonova, Jessica K Hodgins, and Nancy S Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics (ToG)*, 23 (3):514–521, 2004.

Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, April 2009.

Dule Shu, Zijie Li, and Amir Barati Farimani. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, 478:111972, April 2023.

Aliaksandra Shysheya, Cristiana Diaconu, Federico Bergamin, Paris Perdikaris, José M Hernández-Lobato, Richard E Turner, and Emile Mathieu. On conditional diffusion models for pde simulations. *Advances in Neural Information Processing Systems*, 37:23246–23300, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. *Advances in neural information processing systems*, 30, 2017.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302, 2021.

John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31 (2):210–227, 2009.

Pan Xiao, Peijie Qiu, Sung Min Ha, Abdalla Bani, Shuang Zhou, and Aristeidis Sotiras. Sc-vae: Sparse coding-based variational autoencoder with learned ista. *Pattern Recognition*, 161:111187, 2025.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

Yaodong Yu, Sam Buchanan, Druv Pai, Tianzhe Chu, Ziyang Wu, Shengbang Tong, Benjamin Haeffele, and Yi Ma. White-box transformers via sparse rate reduction. *Advances in Neural Information Processing Systems*, 36:9422–9457, 2023.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

A Appendix

B Computational Infrastructure for Experiments

Experiments ran on a single NVIDIA RTX A6000 (48 GB, Driver 535.230.02, CUDA 12.2) under Ubuntu 22.04. Code was executed in PyTorch 2.3.

C Limitation and Discussion

In this paper, we propose the SCDM, a novel framework that integrates structured representation learning into diffusion models to learn the meaningful low-dimensional structures in the reverse diffusion process. Despite its effectiveness, SCDM introduces several hyperparameters, including the lifting dimension, subspace dimension, and compression step size, whose automatic or adaptive selection is left for future work.

D SENSITIVITY ANALYSIS OF BATCH SIZE

To further investigate the effect of $\mathrm{softmax}(Z_t^T Z_t)$, which models pairwise relationships among the n samples within a batch, we conduct a sensitivity experiment on the batch size using the LA dataset. The lifted dimension is fixed to $D = 64$ and the subspace dimension is fixed to $u = 6$. Results are reported over ten random seeds. Stable ratio denotes the percentage of runs that produce valid generations without numerical collapse. For collapsed runs, the generation metrics exhibit abnormally large numerical values, which is consistent with the instability observed under small lifted dimensions in Appendix K. Therefore, these runs are included only for computing the stable ratio and are not included in the reported metric statistics.

Table 4: Sensitivity analysis of batch size. Results are reported over ten random seeds.

Batch Size	**Stable Ratio**	**RMAE** ($\times 10^{-2}$)	**MDN %CE**	**%VFE**
16	20%	25.19 ± 0.12	7.20 ± 0.72	-1.38 ± 1.00
32	50%	25.03 ± 0.30	6.02 ± 2.41	$+0.60 \pm 0.69$
64	90%	25.26 ± 0.41	5.42 ± 2.49	$+0.90 \pm 0.92$
128	100%	25.47 ± 0.56	7.39 ± 2.61	$+2.46 \pm 0.57$
256	100%	25.74 ± 0.54	7.47 ± 3.02	$+3.10 \pm 1.15$

The results show that small batches can still achieve competitive generation metrics. However, their substantially lower stable ratios indicate a much higher probability of numerical collapse. Increasing the batch size mainly improves training stability and robustness.

E ABLATION STUDY OF SCDM

To evaluate the contribution of each component in SCDM, we conduct an ablation study on the LA dataset, whose observations consist of displacement, velocity, and acceleration. As shown in Table 5, we progressively introduce the lifting operation, subspace projection, and structure-aware compression into the vanilla DDPM. The results show that these components provide complementary contributions, with the full SCDM achieving the best RMAE and %VFE.

We further investigate the necessity of the lifting operation by comparing SCDM with and without lifting where the lifting dimension was fixed at $D = 64$. Without lifting, the projected subspace dimension is naturally constrained to be no larger than the original feature dimension. As shown in Table 5, the experimental results are consistent with the observations in Figure 5. When u is restricted to the original input dimension, the comparison with the no-lifting variant still shows the advantage introduced by the lifting operation. Increasing the subspace dimension to $u = 6$ substantially improves the overall generation quality, demonstrating that lifting is important for providing a richer feature space.

Table 5: Ablation study of the main components of SCDM on the LA dataset. All results are reported as mean $\pm$ standard deviation over ten random seeds.

Method	**RMAE** ($\times 10^{-2}$)$\downarrow$	**MDN %CE**$\downarrow$	**%VFE**$\downarrow$
DDPM	25.88 ± 0.76	4.51 ± 2.90	3.41 ± 1.72
DDPM + Lifting	25.54 ± 0.49	7.84 ± 3.24	3.01 ± 1.57
DDPM + Lifting + Projection	25.49 ± 0.93	6.88 ± 3.68	3.58 ± 0.80
SCDM No Lifting ($u = 3$)	25.83 ± 0.54	7.13 ± 2.54	3.78 ± 1.02
Full SCDM ($u = 3$)	25.77 ± 0.44	7.08 ± 3.17	3.85 ± 1.43
Full SCDM ($u = 6$)	25.31 ± 0.90	6.70 ± 3.73	2.60 ± 0.53

Furthermore, we investigate the importance of lifting on CIFAR-10. For a fair comparison, the no-lifting variant directly projects the 256-dimensional U-Net bottleneck features, whereas the full SCDM first lifts them to 512 dimensions; both use the same subspace dimension $u = 32$. As shown

Table 6: Ablation study of the lifting operation on CIFAR-10.

Method	FID↓	IS↑
DDPM	3.17	9.46 ± 0.11
SCDM No Lifting	3.14	9.46 ± 0.10
Full SCDM	2.86	9.95 ± 0.13

in Table 6, the full SCDM achieves better FID and IS, further demonstrating the benefit of lifting for image generation.

F Computational Efficiency Analysis

To evaluate the additional computational overhead introduced by SCDM, we compare the computational efficiency of DDPM, DDPM with only the lifting operation, and the full SCDM on the single linear oscillator (LA) dataset. The experiments are conducted on a single NVIDIA A800 GPU with a batch size of 128. We report the number of parameters, training and sampling throughput, and peak GPU memory consumption.

Table 7: Computational efficiency comparison on the LA dataset. Throughput is reported in samples per second and peak memory consumption in MiB.

Method	Params	Train Thr. (samples/s)	Sample Thr. (samples/s)	Train Mem. (MiB)	Sample Mem. (MiB)
DDPM	1.126M	46,957	377.3	46.36	22.63
DDPM + Lifting	1.193M	35,868	325.6	48.15	22.95
Full SCDM	1.134M	23,723	292.6	46.57	22.70

As shown in Table 7, we acknowledge that the complete structure-aware operation introduces runtime overhead. Nevertheless, the absolute computational cost remains practically manageable for the relatively small physical datasets considered in this word: Full SCDM can still generate approximately 293 samples per second. We consider this overhead acceptable given the improved generation quality and the additional structurally interpretable representation provided bu SCDM.

G Real physical systems

To better demonstrate the generalization performance of SCDM, in this section, we have visualized the first two dimensions of the SCDM sampled in the real world.

Figure 7 demonstrates that SCDM accurately recreates the phase portraits of all six datasets. In DL, synthetic points overlap the tilted rectangular orbit, showing the model captures coupled linear relations. These results confirm SCDM's ability to generalize across heterogeneous physical systems.

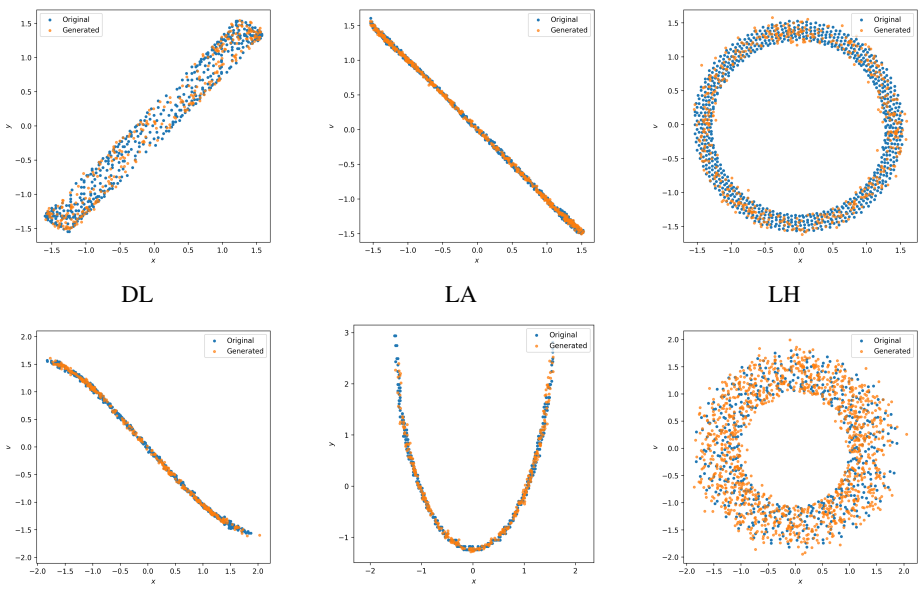
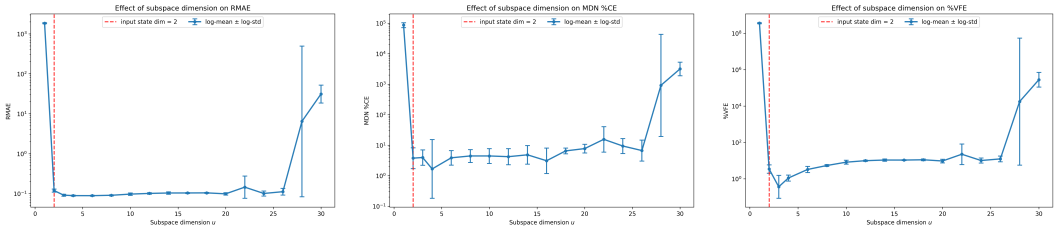


Figure 7: Visualization of the six datasets.

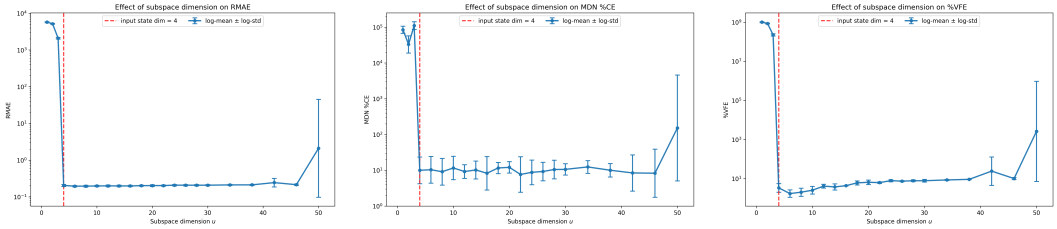
H SENSITIVITY TO SUBSPACE DIMENSION

As shown in Fig. 8 and Fig. 9, the model exhibits a similar trend: when u is too small, the subspace cannot sufficiently cover the underlying state variables, leading to severe degradation in generation quality. After u reaches a moderate range, the performance becomes relatively stable, while excessively large u may weaken the structural constraint and introduce unstable generation behavior.



(a) RMAE (b) MDN %CE (c) %VFE

Figure 8: Effect of the subspace dimension u on the generation quality of the single linear oscillator without acceleration dataset.

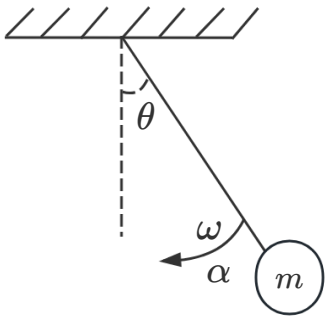


(a) RMAE (b) MDN %CE (c) %VFE

Figure 9: Effect of the subspace dimension u on the generation quality of the double linear oscillator dataset. A similar sensitivity pattern can be observed.

I SUBSPACE STRUCTURAL ANALYSIS

We provide an additional interpretability analysis on the pendulum in angular coordinates (PA). Similar to the LA case, for each learned dimension we use its mean response as a scalar summary and compare it with both raw physical variables and formula-based targets. For PA (θ, ω, α), the identified state quantity is $\alpha = -1.259156\,\sin(\theta) - 0.007308\,\omega + 0.016095$, and the identified energy quantity is $E = \theta^2 + \omega^2$. These targets are used to examine whether the learned subspace representation captures physically meaningful structure beyond directly observed variables. In particular, by comparing the settings with and without explicit acceleration, we find that some learned dimensions remain strongly aligned with the state-related quantity even when α is removed from the observations. This suggests that the representation can still recover acceleration-related state information from (ω, θ) alone.

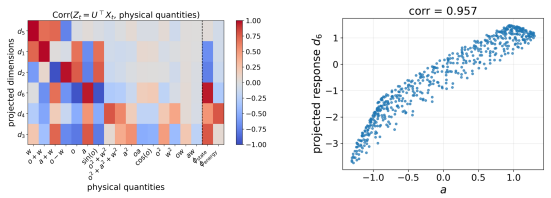


System: pendulum-angular

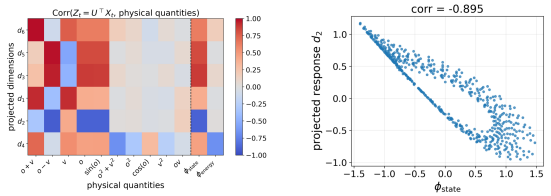
ϕ_{state}:$\alpha = -1.26sin(\theta) - 0.007\omega + 0.02$

ϕ_{energy}: $\omega^2 + \theta^2$

Even without explicit acceleration α in the observations, subspace dimensions can still recover the state quantity from ω and θ.



（c）endulum in angular coordinates with acceleration (θ, ω, α)



（d）endulum in angular coordinates without acceleration(θ, ω)

Figure 10: Interpretability analysis of dimensions in learned low-dimensional subspace for two physical systems: the pendulum in angular coordinates (PA). In each panel, the left part shows the system schematic, the middle part reports the correlation heatmap between dimensions in learned subspace and candidate physical targets, and the right part visualizes the representative relation between a selected dimension and state equation.

J CIFAR10

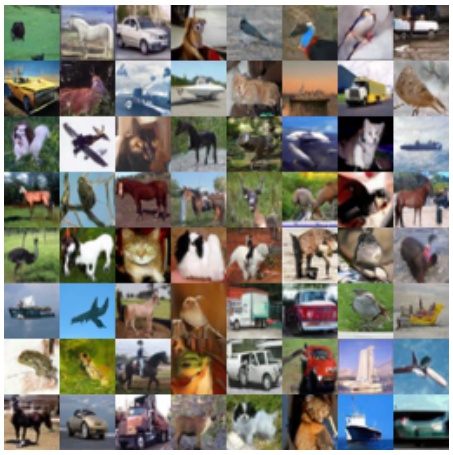
To supplement the quantitative results in the main text, we provide additional CIFAR-10 results in this appendix. Table 8 provides additional quantitative comparisons under severe noisy training data. Under even more severe noisy training conditions, SCDM still shows consistently smaller FID degradation relative to the clean-training setting. Figure 11 shows examples of the noisy training data under 50% corruption, illustrating the severity of the injected Gaussian perturbations. Figure 12 presents samples generated by SCDM under different noisy training settings.

Table 8: FID and IS on CIFAR-10 under 60% noisy training data. The last column reports the FID degradation relative to the clean-training setting (0% noise) for each model.

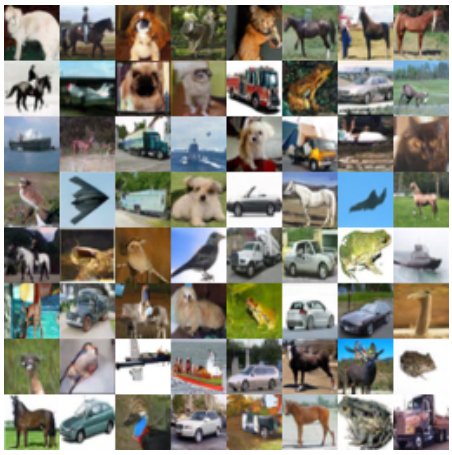
Noise (%)	Model	FID ↓	IS ↑	FID Degradation ↓
60	SCDM	**59.18**	**7.77 ± 0.05**	**56.32**
60	DDPM	68.88	6.96 ± 0.10	65.71
60	SC-VAE	86.18	3.97 ± 0.05	81.77
60	Denoising Diffusion GAN	61.59	7.01 ± 0.11	57.84



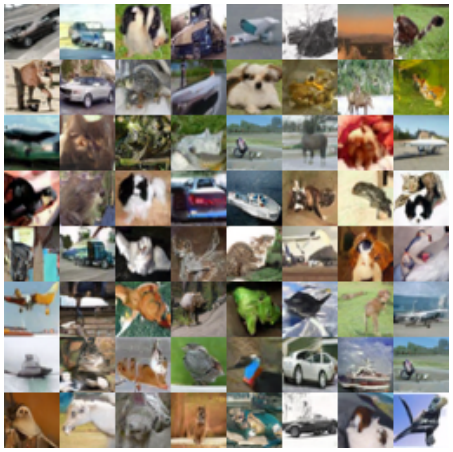
Figure 11: Visualization of the training data under 50% noise corruption on CIFAR10. Odd columns show clean training images, while even columns show their counterparts after adding Gaussian noise with 50% corruption ratio.



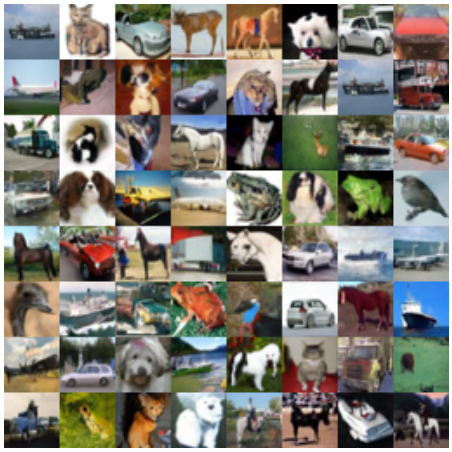
(a) 30% noise, FID=22.75



(b) 40% noise, FID=32.96



(c) 50% noise, FID=51.16



(d) 60% noise, FID=59.18

Figure 12: CIFAR-10 samples generated by SCDM under different noisy training settings.

K Experiment Results for Lifted Dimension on RLA Dataset

In this section, we provide the experiment results for the stability of the lifted dimension on the RLA dataset. For each lifted dimension $D \in \{8, 12, 16, 32, 64, 128\}$, we report the RMAE, MDN %CE, and %VFE values over 30 random seeds. These results complement the summary statistics in Table 2. These results more clearly show that when the lifted dimension D is very small, the model occasionally exhibits unstable runs, where a few random seeds produce collapsed samples and thus extremely large metric values.

Table 9: Results for $D = 8$ and $D = 12$ over 30 random seeds.

Seed	$D = 8$			$D = 12$		
	RMAE ($\times 10^{-2}$)	MDN %CE	%VFE	RMAE ($\times 10^{-2}$)	MDN %CE	%VFE
1	25.38	3.90	-3.97	26.31	4.82	-5.00
2	26.20	2.76	-4.03	25.29	5.62	-4.13
3	2793.29	2771.97	+1913780.82	25.52	1.75	-6.95
4	26.59	2.43	-3.17	26.76	2.25	-3.93
5	25.49	6.99	-1.07	25.50	4.78	-1.25
6	3957.72	3935.91	+2317146.41	25.66	12.80	-6.27
7	26.35	6.37	-0.82	26.48	6.16	-2.06
8	24.91	11.40	-0.48	25.04	11.73	+0.20
9	25.42	15.36	-4.18	25.10	17.20	-1.65
10	25.04	5.17	-6.13	24.55	7.54	-2.79
11	26.26	6.03	+0.95	26.32	4.30	-0.56
12	12511.45	11538.13	+9699816.53	25.97	8.24	-2.96
13	25.12	10.34	-1.96	25.10	13.02	-4.12
14	25.93	11.50	-4.62	25.84	5.05	-0.55
15	999.03	983.81	+217397.74	25.69	11.59	+1.60
16	24.25	4.92	-3.06	24.35	5.33	+1.27
17	24.82	15.61	-5.45	23.85	6.24	-4.32
18	10616.12	10604.87	+11709039.03	25.75	8.05	-1.00
19	25.26	8.03	-2.40	25.26	7.68	+0.93
20	24.75	4.38	-2.16	24.56	0.94	-1.55
21	25.82	0.81	-4.60	25.34	6.64	-2.44
22	1965.68	1908.94	+1465308.38	24.83	5.79	-2.78
23	25.26	8.22	+0.26	24.74	1.38	-0.65
24	26.26	4.55	-2.97	26.29	4.26	+0.65
25	25.53	13.52	-4.47	25.52	11.61	-2.54
26	25.73	9.77	-2.73	25.57	11.02	-2.44
27	26.24	6.86	-2.34	39874.79	39850.33	+2296423441.82
28	26.35	6.31	-1.11	26.16	7.32	-3.70
29	25.31	1.83	-0.97	24.83	4.40	+0.01
30	25.25	5.02	-4.31	25.18	8.20	-1.87

Table 10: Results for $D = 16$ and $D = 32$ over 30 random seeds.

Seed	$D = 16$			$D = 32$		
	RMAE ($\times 10^{-2}$)	MDN %CE	%VFE	RMAE ($\times 10^{-2}$)	MDN %CE	%VFE
1	25.20	6.84	+0.58	24.94	6.94	+1.71
2	25.45	6.49	-4.06	25.53	4.51	-0.24
3	25.04	1.42	-0.63	25.22	0.84	+0.88
4	27.06	2.35	-6.13	27.07	0.90	-4.58
5	25.12	2.88	-2.74	25.04	4.98	+1.61
6	25.51	9.30	-1.45	25.16	8.75	-0.48
7	26.16	6.61	-0.13	26.09	1.74	+1.16
8	25.23	9.01	-1.62	25.23	10.56	+3.91
9	24.92	14.38	+0.43	24.74	14.49	-0.92
10	25.41	6.66	-7.43	24.51	8.50	+1.51
11	26.11	2.48	-0.63	26.28	5.02	+3.69
12	25.53	9.21	-0.75	25.57	6.27	+1.87
13	24.72	9.46	-2.24	24.58	9.69	-0.89
14	25.84	3.95	-2.01	25.70	4.58	+2.08
15	25.22	6.25	+1.68	25.61	7.91	-1.63
16	24.39	4.53	+0.65	24.19	6.13	+0.99
17	23.21	1.54	-2.02	23.41	6.13	-0.98
18	25.40	7.36	-1.28	25.70	10.59	+2.39
19	24.99	6.94	-0.07	25.36	7.39	+1.41
20	24.57	3.46	+1.78	24.61	1.17	+1.80
21	25.39	7.50	-0.71	25.31	5.03	-1.12
22	24.95	4.90	+1.07	24.98	8.07	+1.51
23	24.80	2.45	+2.28	24.70	3.43	+2.38
24	26.43	3.36	-0.59	26.23	2.94	+0.88
25	25.21	7.42	-0.41	25.01	10.34	-0.14
26	25.46	8.79	-2.52	25.31	5.76	+0.15
27	25.81	4.71	-2.16	25.95	5.66	-1.40
28	25.94	7.56	+1.80	25.96	7.09	+1.07
29	24.66	2.82	+0.26	25.06	2.59	-0.15
30	25.07	12.86	-1.46	24.98	12.76	+0.81

Table 11: Results for $D = 64$ and $D = 128$ over 30 random seeds.

Seed	$D = 64$			$D = 128$		
	RMAE ($\times 10^{-2}$)	MDN %CE	%VFE	RMAE ($\times 10^{-2}$)	MDN %CE	%VFE
1	25.17	7.91	+1.61	25.36	6.32	+2.53
2	25.17	5.32	+0.79	25.39	5.74	+0.89
3	25.06	3.04	+2.18	24.97	2.55	+0.80
4	26.55	1.84	+0.24	26.13	1.94	+1.63
5	25.48	4.13	+0.96	25.19	4.61	+1.76
6	25.19	9.37	+0.70	25.28	6.77	-0.32
7	26.10	3.79	+2.24	26.01	2.49	+1.92
8	24.92	8.95	+3.47	25.00	9.15	+2.80
9	24.80	14.04	-0.28	24.76	16.38	+0.32
10	24.51	9.55	+1.42	24.60	11.89	+2.08
11	26.10	3.91	+3.49	26.03	4.49	+2.64
12	25.65	4.48	+1.95	25.68	5.17	+1.93
13	24.97	16.07	-0.36	24.58	13.13	+0.95
14	25.82	3.19	+1.61	25.50	4.10	+0.15
15	25.24	8.00	+3.39	25.38	7.12	+2.92
16	24.10	5.73	+2.07	24.28	7.14	+2.36
17	23.28	6.21	-1.59	23.08	2.90	+0.22
18	25.68	10.11	+1.85	25.70	9.04	+2.60
19	25.15	5.97	+2.07	25.11	7.18	+1.33
20	24.83	1.47	+2.42	24.33	1.83	+1.80
21	25.37	6.45	+0.19	25.38	6.06	+0.59
22	24.79	8.85	+1.15	24.77	8.47	+0.58
23	24.66	4.34	+1.42	24.89	1.57	-1.37
24	26.58	5.33	+0.95	26.14	4.37	+1.78
25	24.90	6.48	+0.78	24.97	5.78	+1.56
26	25.40	9.35	-0.25	25.30	4.47	+0.60
27	25.84	5.78	+1.73	25.46	5.70	+1.63
28	25.83	5.55	+2.14	25.92	4.94	+2.69
29	24.66	4.44	+0.93	24.71	3.08	+1.90
30	24.74	11.90	+1.65	24.77	9.59	-0.64