BioDM: Bio-signal-guided Denoising Diffusion Probabilistic Model for Biological Dynamics

Anonymous submission

Abstract

Biological dynamics are fundamental in uncovering mechanisms, yet their intricate nature and the substantial expense of data acquisition make accurate capture a significant challenge. While generative models like Denoising Diffusion Probabilistic Models (DDPMs) have shown promise in generating advanced texts and images, their application to biological dynamics faces several key limitations: **(1) Limited Global Principle-based Semantics:** While biological images implicitly contain crucial semantics governing biological processes, these are difficult for DDPMs to learn due to the lack of explicit representation of key variables like cell areas and the model's focus on distribution approximation rather than capturing underlying biological mechanisms. **(2) Entanglement of Causal and Non-Causal Effects:** DDPMs struggle to disentangle causal and non-causal effects in biological images due to the denoising process, hindering interpretability and limiting their utility for downstream biological tasks. **(3) Lack of Large-Scale Datasets:** The absence of comprehensive biological datasets encompassing dynamics further hinders progress in this area. To address these challenges, we propose *bio-signal-guided denoising diffusion probabilistic model* (*BioDM*), which leverages the Information Bottleneck (IB) mechanism to capture biological dynamics. The IB mechanism decomposes DDPM, enabling it to: *(1) Capture Global Principle-based Semantics: BioDM* effectively captures global semantics and underlying principles governing biological dynamics. *(2) Focus on Causal Effects: BioDM* guides DDPM to focus on causal relationships crucial for understanding biological dynamics. *(3) Contribution of Large-Scale Datasets (BioDBench):* Furthermore, we contribute the large-scale biological dynamics dataset (BioDBench), comprising two single-cell datasets and a jellyfish-robot dataset. Extensive experiments on these datasets show the superiority of *BioDM* in capturing causal effects, advancing understanding of biological dynamics.

Introduction

The dynamic nature of biological systems, characterized by intricate interactions and complex processes, lies at the heart of biological research. Understanding these dynamics is crucial for elucidating the mechanisms underlying biological phenomena, spanning from molecular interactions (de Haan and Rottier 2005; Southern, Mir, and Shchepinov 1999) to organismal behavior (Diogo 2017; Wcislo 2021). Accurate and comprehensive capture of these dynamics is essential for advancing biological discoveries, but remains a significant challenge due to the inherent complexity of biological systems and the cost associated with data acquisition.

While significant efforts have been made to explore biological dynamics, with recent work focusing on generative techniques like TVAE for RNA sequence deconvolution in cells (Nishikawa, Lee, and Amau 2024) and investigations into protein expression and adaptive immune receptors (Heumos et al. 2023), there remains a critical need to delve deeper into biological dynamics at the cellular level. Understanding cellular dynamics offers a powerful lens for global-scale analysis, providing biologists with insights into the fundamental processes driving biological systems. Furthermore, exploring these dynamics at the cellular level holds immense potential for predicting biological processes ahead of time, enabling more targeted interventions and facilitating the development of therapeutic strategies.

While generative models, particularly Denoising Diffusion Probabilistic Models (DDPMs) (Ho, Jain, and Abbeel 2020), have obtained great success in many domains such as natural language processing (NLP) (Zhu and Zhao 2023; Zou, Kim, and Kang 2023) and computer vision (CV) (Po et al. 2024; Yan, Gu, and Rush 2024; Peebles and Xie 2023), their applications to biological dynamics face significant limitations. **(1) Limited Global Principle-based Semantics:** These semantics are implicitly embedded in the biological images and also hard for DDPMs to learn, yet they are crucial for governing biological processes. On one hand, many important variables of bio-signal such as cell areas during their life cycles, are not explicitly represented in the input. On the other hand, DDPMs primarily focus on generating the target distribution based on prior time points' sample, approximating the generated samples' distributions to that of ground truth. **(2) Entanglement of Causal and Non-Causal Effects:** DDPMs frequently entangle causal and non-causal effects of the biological image, hindering interpretability and limiting their utility for downstream biological tasks. This entanglement arises from generating biological images through denoising, potentially leading to a conflation of bio-signal related and pure noisy factors. However, only part of the pixels are causal for the simulation of the biological process. **(3) Lack of Large-Scale Datasets:** The scarcity of large-scale biological datasets encompassing dynamic processes further exacerbates these challenges. The

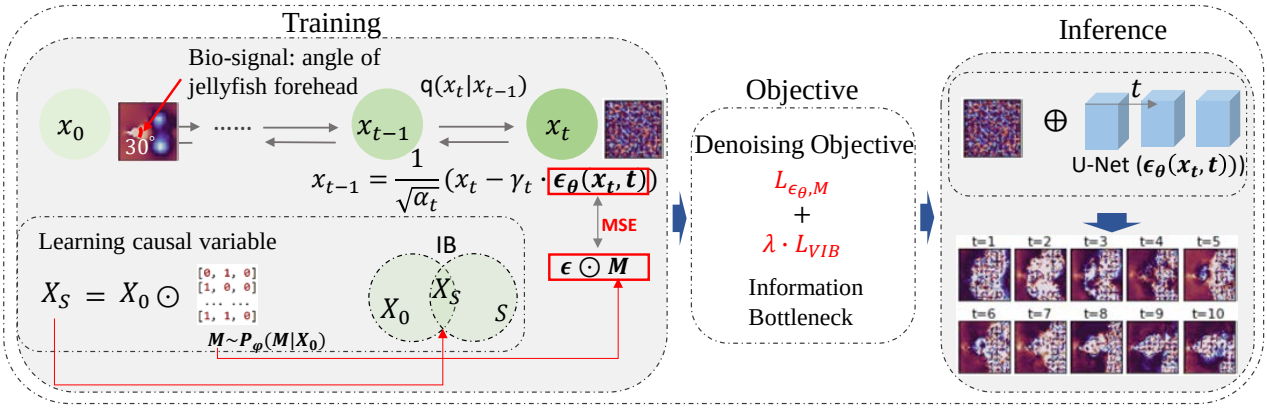


Figure 1: Architecture of our proposed ***BioDM*** framework. Our method consists of three main components. The first component involves the training process using the Information Bottleneck (IB) mechanism, which is constrained by the objective function (described in the second component). The third component illustrates the inference process.

absence of comprehensive datasets hinders the development and evaluation of models capable of accurately capturing the complex dynamics of biological systems.

To address the above limitations, we introduce a novel approach called the *bio-signal-guided denoising diffusion probabilistic model* (*BioDM*). We propose to capture bio-signals through a convolution layer and incorporate global principle-based semantics into *BioDM* as structured prior information. This enhancement amplifies the explicit content of the biological dynamics observed by the algorithm, enabling it to swiftly navigate toward the correct generation and exhibit strong generalization capabilities. In addition, *BioDM* designs the auxiliary latent mask guided by the principles of the Information Bottleneck (Tishby Naftali and William 2000; Tishby and Zaslavsky 2015). Through paying more attention to causal features related to bio-signal, *BioDM* generates discriminative and interpretable dynamics consistent with the real biological process. To further support the development and evaluation of simulating biological dynamics, we have compiled a comprehensive biological dataset (*BioDBench*). This dataset includes two single-cell subsets and one jellyfish-robot subset, each featuring *10K videos* (equivalent to *200K images*) accompanied by bio-signals describing the global principles.

In summary, we make the following contributions:

- **Novel Paradigm for Capturing Biological Dynamics.** We propose a novel paradigm, built upon the DDPM model, for capturing biological dynamics. This approach, guided by bio-signals, represents a novel endeavor in capturing biological dynamics, offering valuable insights for the research community.
- ***BioDM* with Information Bottleneck Decomposition.** We propose **the Biosingal-guided denoising diffusion probabilistic model (*BioDM*)** decomposed by the Information Bottleneck (IB) mechanism to improve the capture of biological dynamics, including **Global Principle-based Semantics Capture** and **Focus on Causal Effects**. These advancements empower *BioDM* to generate more accurate and interpretable representations of biological dynamics, facilitating deeper insights into complex biological systems.
- **Data Contributions and Extensive Experiments.** We introduce *BioDM*, a novel diffusion-based model for bio-signal-guided generation of biological dynamics. We contribute a large-scale dataset, *BioDBench*, providing a rich source of data for training and evaluation. Extensive experiments on these datasets demonstrate that *BioDM* outperforms seven state-of-the-art baselines across all metrics. The code for reproducing these results is available at https://anonymous.4open.science/r/BioDM-6B05/README.md. We also attached reproducibility checklist.

Related Work

Causal Relationships Mining for Artificial Intelligence. Causal relationships mining, also known as attribution discovery, is an ever-evolving research field aimed to understand the significance and contribution of different input features (Springenberg et al. 2014; Ribeiro, Singh, and Guestrin 2016; Selvaraju et al. 2017). For example, Gradient (Baehrens et al. 2010) and Saliency (Simonyan and Zisserman 2014) compute the gradient of the target output neuron in relation to the input features. SmoothGrad (Smilkov et al. 2017) enhances gradient-based attribution maps by averaging gradients across multiple inputs, employing techniques such as brightness level interpolations or considering a local neighborhood. In a manner similar to our work, MacDonald et al. (2019) adopts a rate-distortion perspective; however, their focus is on minimizing the norm of the mask rather than emphasizing shared information. The goal of (Schulz et al. 2020) is to incorporate the information bottleneck as an explanatory component to shed light on fixed and trained neural networks using standard stochastic gradient methods. These attribution techniques like (Schulz et al. 2020) necessitate that the task at hand involves classification and their objective is to associate the decision made for a particular class with the significance of pixels in a provided dataset. Different from these studies, our research is to estimate the amount of information utilized for focus on casual effect in diffusion models to explore biological dynamics.

AI for Biological Science. With the success of deep learning techniques in medical CV (Chen et al. 2022; Singh et al. 2020), many researchers have begun to explore biology sci-

ence like gene network prediction in (Kalfon et al. 2024) and generative techniques like LLMs for scientific discovery (AI4Science and Quantum 2023). To be specific, Toui *et al.* (Nishikawa, Lee, and Amau 2024) investigates a generative method via TVAE to improve the bulk deconvolution.Another recent work on single-cell trajectory inference (Van den Berge et al. 2020) proposed a novel generalized additive model framework that enables flexible inference of both within-lineage and between-lineage differential expression, utilizing the negative binomial distribution. This framework further incorporates observation-level weights to account for zero inflation, enhancing its robustness and applicability. However, few of existing studies to explore the biological dynamics at the cell level. Motivated the generative technique's success on biology science, we aim to explore a bio-signal-guided DDPM method decomposed with the Information Bottleneck (IB) mechanism to investigate biological dynamics. We also provide other contexts of related work in Appendix A.3.

Method

This section presents the technical details of our proposed method, *BioDM*, as illustrated in Figure 1. We begin by formally defining the problem we aim to address. Subsequently, we provide a detailed exposition of *BioDM*, encompassing a concise overview of Denoising Diffusion Probabilistic Models (DDPMs), the architectural design of *BioDM*, and a brief analysis of its learning efficiency.

Problem definition

Given the high-dimensional biological dynamics data $X = \{x^{(n)}\}_{n=1}^{N} \in \mathbb{R}^D$ [1] Accompanying each $x^{(n)}$ is the low-dimensional biological signal $s^{(n)} \in \mathbb{R}^d$ for signal variable S. Here, instead of the standard task of learning the distribution[2] $P_X(x)$ for generating X, we are interested in learning a distribution $P_W(w)$ for generating the *causal* variable W, such that W is as faithful to X as possible while containing exclusively *causal* features to S. Concretely, we want to learn a distribution $P_W(w)$ for sampling W, where $W = g(X, S) \in R^D$ has the same dimension as X, such that the *causal* features of W w.r.t. S obey the same distribution as X and the non-S-causal features obey a Gaussian distribution. Note that during inference time, we do not have the accompanying signal S for generating W. Thus, it is impossible to first generate X and then use attribution methods to find which features of X are relevant to S and then produce W. Please refer to Figure 2. Therefore, the model for generating W must be learned at training time.

R#J6Qk-W1, R#Ai1S-Q1, R#4aBP-Q1, R#xZiv-W2

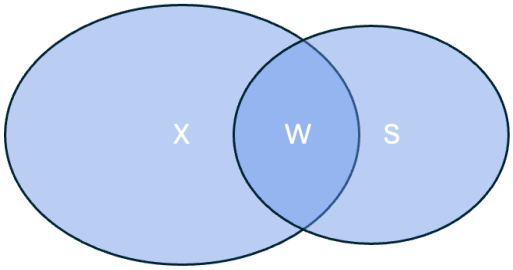


Figure 2: Information diagram for the biological dynamics data X, bio-signal S, and the causal variable W. Instead of standard DDPM which generates X, our BioDM aims to generate W which is contained in X and relevant to the bio-signal S, i.e. lying on the mutual information part $I(X; S)$.

R#J6Qk-W1, R#Ai1S-Q1, R#4aBP-Q1, R#xZiv-W2

Bio-signal-guided Denoising Diffusion Probabilistic Models (*BioDM*)

The DDPM method. To tackle the above task, we build upon the recent advances of Denoising Diffusion Probabilistic Models (DDPMs) (Ho, Jain, and Abbeel 2020). DDPM is an elegant method to learn high-dimensional distributions $P_X(x)$ given data samples $\{x^{(n)}\}_{n=1}^{N}$. Concretely, DDPM consists of a forward process that adds t steps of Gaussian noise to the data sample[3] x_0 to obtain a noisy sample x_t: $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ has the same dimension as x_0 and $\bar{\alpha}_t$ is a predefined schedule. Since the summation of Gaussian is also a Gaussian, the t steps of adding Gaussian noise is equivalent to adding a single Gaussian $\sqrt{1-\bar{\alpha}_t}\epsilon$. After T steps of forward process, x_T approximates a standard Gaussian distribution. In the reverse process, the DDPM learns a denoising model $\epsilon_\theta(x_t, t)$ that aims to revert the forward process:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\sqrt{1-\alpha_t}}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t\eta, t = T, ...1.$$

Here α_t, σ_t are pre-defined schedule and $\eta \sim \mathcal{N}(0, I)$. To learn $\epsilon_\theta(x_t, t)$, DDPM use the denoising objective

$$L_{\epsilon_\theta} = ||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||_2^2, \epsilon \sim \mathcal{N}(0, I) \quad (1)$$

which aims to predict the added noise ϵ based on the noisy sample $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$. For more details about DDPM, see Appendix A.4.

The bio-signal-guided DDPM method (*BioDM*). Although the above DDPM can learn complicated distribution $P_X(x)$ given data samples, it is insufficient to learn the distribution $P_W(w)$ for the causal variable W since there are no ground-truth data W to learn the distribution from. Our key insight is that the relevance discovery of X to the signal

[1] Here we use capital letters (e.g., X) to denote random variables and lowercase letters (e.g., x) to denote their instances. The lowercase letters with superscript (e.g., $x^{(n)}$) denote data samples.

[2] Sometimes, we may be interested in learning a condition distribution $P_{X|C}(x|c)$ for optional condition variable C. For notation simplicity, we ignore such conditions and only add it as needed.

[3] Here the subscript t in x_t denotes the denoising step.

Algorithm 1: ***BioDM*** training

1: **repeat**
2: $x_0 \sim p(x_0)$
3: $\epsilon \sim \mathcal{N}(0, I)$
4: Take gradient descent step of L_{BioDM} w.r.t. θ and φ:

$$\nabla_{(\theta,\varphi)}[||\epsilon \odot M - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||_2^2 + \lambda L_{\text{VIB}}]$$

5: **until** converged

Table 1: Data Statistics

Datasets	Tension		Wetting		Fluid (Jellyfish)	
Types	Images	Videos	Images	Videos	Images	Videos
Number	100,000	5,000	100,000	5,000	10,000	500
Time Steps	-	20	-	20	-	20
Resolution	256 × 256	-	256 × 256	-	256 × 256	-
Size	~500G		~500G		~10G	
Summary	~1T					

S is essentially finding the *minimal sufficient* information contained within X for predicting S, and the Information Bottleneck (Tishby Naftali and William 2000) provides the exact principle and technique we need for extracting such minimal sufficient information. Specifically, we consider a noisy representation X_S of X where X_S only contains the relevant features of X w.r.t. S. To learn such a representation, we employ the following Information Bottleneck (IB) objective:

$$L_{\text{IB}} = I(X; X_S) - \beta \cdot I(X_S; S) \tag{2}$$

Here $I(\cdot;\cdot)$ denotes mutual information, and β is a hyperparameter. By minimizing the above IB objective, it encourages X_S to contain as much information as possible for predicting S, while retaining as much little information as possible about X, thus encouraging X_S to contain the minimal sufficient information of X for predicting S.

We define X_S as X multiplied with a continuous mask $M \in [0,1]^D$, where the mask values indicate the per-feature relevance:

$$X_S = X \odot M \tag{3}$$

Here $\odot$ denotes element-wise multiplication. To obtain M, we use an encoder network $p_\varphi(M|X)$ with learnable parameters φ. Since the IB objective is intractable, we employ the deep Variational Information Bottleneck (VIB) (Alemi et al. 2016) to minimize its upper bound:

$$L_{\text{VIB}} = \mathbb{E}_{X_S \sim p_\varphi(X_S|X)}\left[\log \frac{p_\varphi(X_S|X)}{q_\varphi(X_S)} - \beta \log q_\varphi(S|X_S)\right] \tag{4}$$

The first term provides an upper bound for $I(X; X_S)$ and the second term provides an upper bound for $-\beta \cdot I(X_S; S)$ in Eq. 2. Here the encoder $p_\varphi(X_S|X)$, the prior distribution $q_\varphi(X_S)$, and the decoder $q_\varphi(S|X_S)$ are all parameterized by learnable neural networks. Minimizing the above L_{VIB} objective encourages finding an encoder $p_\varphi(M|X)$ for the mask M so that the $X_S = X \odot M$ approximately extracts the minimal necessary information of X for S.

Given the above IB module, how can we encourage the DDPM to learn to generate the causal variable W relevant to S? Note that the objective L_{ϵ_θ} in Eq. 1 encourages the denoising network ϵ_θ to denoise all features of x_t. Empowered with the mask M found by IB, our ***BioDM*** method instead *only* denoise the features deemed relevant by M. Concretely, we introduce the following modified denoising objective:

$$L_{\epsilon_\theta, M} = ||\epsilon \odot M - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||_2^2, \epsilon \sim \mathcal{N}(0, I) \tag{5}$$

Here $M \sim q_\varphi(M|X)$ is obtained by the VIB. For ***BioDM***, the above denoising objective and the VIB objective are jointly optimized, constituting the full ***BioDM*** objective:

$$\begin{aligned} L_{\text{BioDM}} &= L_{\epsilon_\theta, M} + \lambda \cdot L_{\text{VIB}} \\ &= \mathbb{E}\Bigg[||\epsilon \odot M - \epsilon_\theta(\sqrt{\bar{\alpha}_t}X + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||_2^2 \\ &\quad + \lambda \cdot \big(\log \frac{p_\varphi(X_S|X)}{q_\varphi(X_S)} - \beta \log q_\varphi(S|X_S)\big)\Bigg] \end{aligned} \tag{6}$$

Here the expectation is taken w.r.t. $(X, S) \sim p(X, S)$ (from data), $X_S \sim p_\varphi(X_S|X)$, and $\epsilon \sim \mathcal{N}(0, I)$. During training, the VIB objective L_{VIB} helps discover the mask M that indicates the relevant features of X to S. For features that are found relevant, the corresponding mask elements in M are near 1, so the ***BioDM***'s denoising objective $L_{\epsilon_\theta, M}$ reverts to the DDPM objective L_{ϵ_θ}. On the other hand, for the features that are deemed irrelevant to S, the corresponding mask elements in M are approximately 0, and the corresponding features in the noise target $\epsilon \odot M$ are 0. In other words, the $L_{\epsilon_\theta, M}$ trains ϵ_θ to predict zero noise on these irrelevant features. At inference time, ***BioDM*** generates the causal variable W using the same procedure as DDPM:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\sqrt{1-\alpha_t}}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t \eta, t = T, ...1. \tag{7}$$

starting with $x_T \sim \mathcal{N}(0, I)$, and we have $W := x_0$ as the last sample. We see that at inference, the irrelevant features are not denoised and remain Gaussian because the ϵ_θ is trained to predict 0 for these features. In this way, we can generate the causal variable W in inference time where only the relevant features of S are denoised.

Taken together, the modified denoising objective $L_{\epsilon_\theta, M}$ (Eq. 5) and the IB module for discovering the relevant features (Eq. 4) constitute the key innovation of ***BioDM***. We provide the algorithm for training as Alg. 1.

Architecture design. Here we detail the neural network architecture for ***BioDM***. We use grid-based data (e.g., image and videos) to illustrate the architectural design[4]. For the denoising network ϵ_θ, we use the standard choice of U-Net (Ronneberger, Fischer, and Brox 2015). For modeling the distribution $p_\varphi(X_S|X)$, we first use a U-Net that takes as input X and returns the estimation of mean $\mu_{M,\varphi}(X)$ and logit $\xi_{M,\varphi}(X)$ for the mask M as its feature

[4]Note that our method is fully general and can also deal with other types of input such as graph and sequence.

Table 2: We comprehensively evaluate our model against baselines on Wetting, Tension, and Fluid datasets, demonstrating superior performance across all tasks. Our model excels at generating dynamic videos with relevant signal pixels while maintaining noise in irrelevant areas, and accurately identifying pressure field components relevant to boundary force measurements in fluid simulations.

	Wetting				Tension				Fluid	
	Area		Circumference		Area		Circumference		Force	
Method	IoU	Sensitivity-n	IoU	Sensitivity-n	IoU	Sensitivity-n	IoU	Sensitivity-n	Correlation	Sensitivity-n
Random	0.2389	0.0315	0.0717	0.1338	0.3297	0.0094	0.1744	0.0467	0.0003	0.2432
Gradient	0.2147	0.0181	0.0882	0.0153	0.5309	0.0015	0.2636	0.0044	0.0064	0.0051
Saliency	0.2811	0.0290	0.0892	0.1252	0.4896	0.0151	0.2414	0.0308	0.0327	0.1806
GuidedBP	0.1033	0.0734	0.0238	0.1065	0.0350	0.0368	0.1399	0.0111	0.0033	0.2021
GuidedCAM	0.0601	0.0199	0.0221	0.0980	0.0399	0.0081	0.0734	0.0085	0.0085	0.1974
SmoothGrad	0.1755	0.0049	0.0612	0.1677	0.4691	0.0211	0.3503	0.0451	0.0093	0.0229
GuidedGrad	0.0723	0.0290	0.0350	0.1017	0.1272	0.0114	0.0514	0.0392	0.0285	0.2182
Custom Diffusion	0.3236	0.0607	0.0885	0.0101	0.0756	0.0160	0.2780	0.0768	0.0027	0.0226
DAAM	0.2203	0.0075	0.0966	0.0269	0.0779	0.0311	0.0468	0.0037	0.0056	0.0422
BioDM (ours)	**0.4590**	**0.3114**	**0.1197**	**0.1713**	**0.6689**	**0.1638**	**0.3699**	**0.1312**	**0.3343**	**0.3881**

maps. Then we use the reparameterization trick (Kingma and Welling 2013) to represent $p_\varphi(X_S|X)$ with the Gaussian $\mathcal{N}(X_S; \mu_{X_S,\varphi}(X), \sigma^2_{X_S,\varphi}(X))$, where the mean and the variance are conditioned on X as follows:

$$\mu_{X_S,\varphi}(X) = \text{clamp}_{[0,1]}(\mu_{M,\varphi}(X)) \odot X$$
$$\sigma^2_{X_S,\varphi}(X) = \text{softplus}(\xi_{M,\varphi}(X) \odot X)$$

We employ $\text{clamp}_{[0,1]}$ to make sure that the mask mean stays between 0 and 1, and $\text{softplus}(x) = \log(1 + e^x)$ to ensure that the variance is non-negative. For the prior term $q_\varphi(X_S)$ in L_{VIB} (Eq. 4), there are multiple options. The most general way is to use a mixture of full Gaussians where the mixing weight and the Gaussian mean and covariance matrix are learnable. Instead, we find that the simplest choice of letting $q_\varphi(X_S)$ be a diagonal Gaussian $\mathcal{N}(X_S; 0, I)$ works quite well. For $q_\varphi(S|X_S)$, we assume that S obeys a diagonal Gaussian $\mathcal{N}(S; \mu_{S,\varphi}(X_S), I)$ where $\mu_{S,\varphi}(X_S)$ can be a U-Net encoder followed by a Multilayer Perceptron (MLP). Thus, $\log q_\varphi(S|X_S)$ reduces to a standard MSE loss. Combined together, L_{VIB} in Eq. 4 reduces to

$$L_{\text{VIB}} = \mathbb{E}_{(X,S)\sim p(X,S)}\Bigg[\frac{1}{2}\beta||\mu_{S,\varphi}(X_S) - S||_2^2 + \frac{1}{2}\big(1 + \sum_{j=1}^{D}\big(\log\sigma^2_{X_S,\varphi}(X) - \mu^2_{X_S,\varphi}(X) - \sigma^2_{X_S,\varphi}(X)\big)_j\big)\Bigg] \tag{8}$$

Here in the second term, the j denotes the jth feature for the variable, and it sums over D dimensions since $X_S \in R^D$. For detailed derivation, see Appendix A.1.

Remark on learning speed. Since *BioDM* only needs to learn to denoise the relevant features which contain much less information than the full variable, it is much more efficient to train than the original DDPM. Empirically, we find that *BioDM* typically converges around twice as fast as the DDPM method. This demonstrates the added benefits of how *BioDM* for improving learning speed.

Experiments

In this section, we aim to answer the following questions: **RQ1:** How is the explanation ability of our method *BioDM* compared with other state-of-the-art methods (*e.g.*, SmoothGrad and GuidedGrad-CAM) on three datasets (wetting, tension and fluid) in terms of three? **RQ2:** How does each component of our method *BioDM* affect its explanation capability? **RQ3:** What is the efficiency of our method *BioDM* compared with other baselines?

Experimental Setup

Datasets. Our dataset *BioDBench* include two cell dynamics datasets and a fluid dynamics dataset. Detailed information about these datasets, including specific statistics, can be found in Table 1. The wetting dataset explores morphological changes of a system (such as cells) under various scenarios involving grid and domain configurations, tension, adhesion, and other physical parameters. It is generated through simulations with systematically modified key parameters to observe wetting behavior. In contrast, the tension dataset examines how the shape of a system evolves under specific grid and domain conditions, achieved by adjusting parameters like tension parameters. As an example, in Figure 5 (b), each trajectory X consists of cell states across 20 time steps. We see that the cell shape gradually becomes more rounded as time progresses within each trajectory due to the surface tension. In addition, we have accompanying signal S of cell areas or circumferences. Our focus extends beyond learning a probabilistic model for generating cell trajectory videos. We are equally interested in identifying the which video pixels corresponds to the signal of area or circumferences. For the Fluid dataset, please refer to Appendix A.2 for a detailed explanation. As exemplified in Figure 5 (e), our interest lies not only in predicting the pressure field conditioned on the boundary but also in discerning which components of the pressure field contribute to the force measurements acting on the boundary. We also provide detailed description of Evaluation metrics in Appendix A.5.

Baselines. We performed comprehensive experiments, comparing our method *BioDM* with several state-of-the-art base-

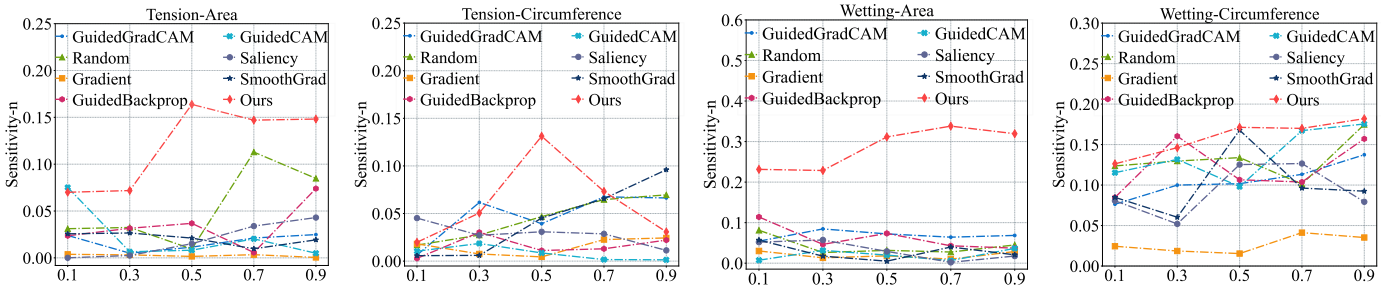


Figure 3: The Sensitivity-n values of all methods were evaluated on two datasets, specifically tension and wetting. The range of masked pixels varied from 10% to 90%. The definition of Sensitivity-n can be found in Eq. 17. It was observed that the Sensitivity-n values did not significantly differentiate between the baseline methods. However, our proposed method *BioDM* outperformed the others for most masking percentages, demonstrating that our method can discover relevant pixels that are highly correlated to the prediction of the signal.

R#xZiv-W2

lines on three distinct datasets. These baselines include Random (Schulz et al. 2020), Gradient (Baehrens et al. 2010), Saliency (Simonyan and Zisserman 2014), GuidedBP (Springenberg et al. 2014), GuidedCAM (Selvaraju et al. 2017), SmoothGrad (Smilkov et al. 2017), and GuidedGrad (Ribeiro, Singh, and Guestrin 2016), Custom Diffusion (Wang et al. 2023), DAAM (Tang et al. 2022). To help reproduce the experiment results of all methods, we also provide implementations of baselines in Appendix A.6.

Overall Comparison (RQ1)

We evaluate all methods using three metrics: IoU, Sensitivity-n (50% pixel masking), and Correlation (Table 2). Visual results of state-of-the-art baselines and our proposed *BioDM* are presented in Figures 5 (More visualization results in Figure 6, Figure 7 and Figure 8 are shown in Appendix A.6). Additional experiments involve masking pixels in generated samples to assess explanation capability (Figure 3). Sensitivity-n average values are provided in Table 4 (Appendix A.6). From these results, we make the following observations:

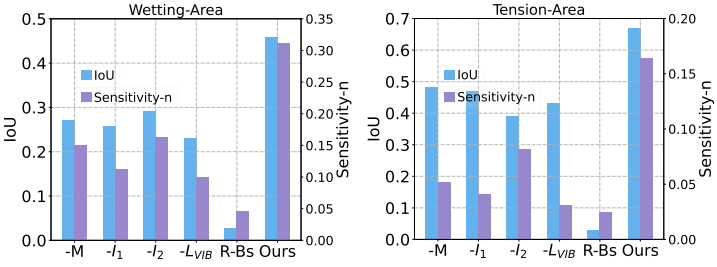


Figure 4: Ablation Study

Superior Performance. Based on the comprehensive analysis presented in Table 2, our innovative methodology demonstrates consistent superiority over competing approaches across all evaluation metrics on the three diverse datasets examined. This exceptional performance can be attributed to a set of reasons that distinguish our model, *BioDM*, from baselines, shown as follows:

IB Improves Capturing of Casual Effects: The success of *BioDM* stems from its innovative decomposition with the Information Bottleneck mechanism. The Information Bottleneck (IB) mechanism strategically extracts relevant features causing the signal variable, enhancing the causal effects between generated data and key characteristics.

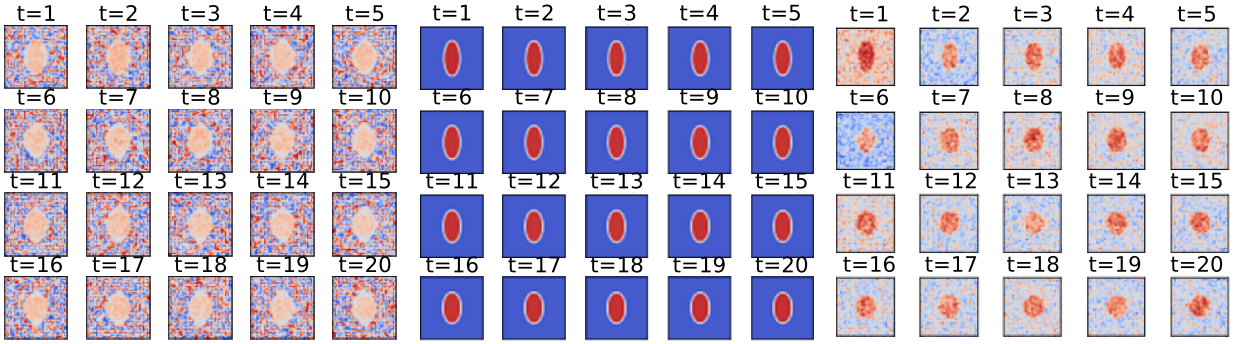
Impact of Mask Mechanism: The novel mask mechanism refines the fidelity of generated samples by targeting Gaussian noise components, ensuring alignment with actual data and improving output quality. Gradient-based techniques like Gradient, Saliency, and SmoothGrad effectively capture and accentuate pertinent features, consistently outperforming across datasets. This holistic approach combines feature discovery, mask mechanism, and gradient-based analysis to create a robust and versatile model that excels in data-driven tasks.

More Relevant Generation to Bio-signals. In Figure 3, we present the Sensitivity-n values obtained by all methods on images from the wetting and tension datasets. We see that our method, *BioDM*, gives a much higher Sensitivity-n values than the other baselines in most conditions (masking percentage). This shows that our method *BioDM* has the strongest ability to identify relevant pixels that are highly correlated to biological signal.

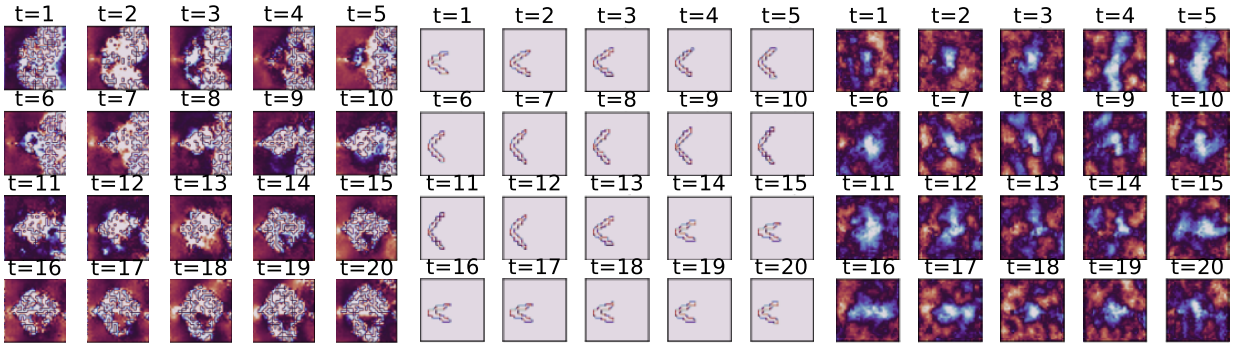
Visualization on Tension and Fluid Datasets. Figures 5 showcase the generated samples derived from the best baseline, Gradient, alongside our proposed approach. Within Figure 5, the amalgamation of colors signifies areas deemed *non-relevant* to the signal parameters such as cell areas, circumferences, and water pressures, as discerned by the respective methodologies. Noteworthy observations include:

Better Ability to Generate Relevant Components to Bio-Signals. Both our method and Gradient exhibit proficiency in generating samples featuring pertinent components related to cell areas; however, our method excels in enhancing relevance and explanatory capacity. We attribute this to decomposition of the IB mechanism, helping capture causal effects in biological datasets.

Better Effectiveness on the Hard Dataset (Fluid). As illustrated in Figure 5, our method demonstrates a notable ability to identify critical segments of the pressure field, particularly at the jellyfish's leading edge, which directly influence boundary forces. This capability is not observed in the leading baseline method, highlighting the efficacy and broad applicability of our innovative approach. This result verifies the effectiveness of integrating bio-signals within our model, *BioDM*, enabling it to capture global, principle-



(a) **BioDM (ours)** (Tension) (b) Ground Truth (Tension) (c) Gradient (Tension)



(d) **BioDM (ours)** (Fluid) (e) Boundary (Fluid) (f) Gradient (Fluid)

R#xZiv-W2

Figure 5: **For the first line:**Visualizations on the Tension dataset demonstrate that *BioDM* generates more accurate and consistent cell area representations compared to the Gradient baseline, highlighting its superior causal ability in capturing cell dynamics.**For the second line:** Visualizations on the Fluid dataset highlight *BioDM*'s superior ability to accurately identify relevant pressure field components for boundary force measurement compared to the Gradient baseline, which struggles to generate consistent fields and misidentifies relevant pixels.

based semantics (i.e., dynamic water pressure) inherent in biological datasets.

Table 3: Training time (in hours) is reported for each method on three datasets.

Methods	Tension		Wetting		Fluid	
	Time (h)	IoU	Time (h)	IoU	Time (h)	Correlation
Random	2.16	0.2398	2.13	0.3297	2.21	0.0003
Gradient	4.83	0.2147	5.33	0.5309	5.33	0.0064
Saliency	5.18	0.2811	4.83	0.4896	5.33	0.0327
GuidedBP	5.33	0.1033	5.33	0.30350	5.25	0.0033
GuidedCAM	5.16	0.0601	5.16	0.0399	6.25	0.0085
SmoothGrad	11.08	0.1755	10.33	0.4691	12.16	0.0093
GuidedGrad	8.55	0.0723	8.67	0.1272	9.15	0.0285
Custom Diffusion	4.30	0.3236	4.52	0.0756	5.16	0.0027
DAAM	4.34	0.2203	4.51	0.0779	5.61	0.0056
BioDM (ours)	2.35	**0.4590**	2.27	**0.6689**	8.19	**0.3343**

Ablation Study (RQ2)

We orchestrated a series of experiments aimed at dissecting the individual contributions of each element within our methodology, as delineated in the enlightening Figure 4. The quartet of variants scrutinized encompass: (1) "-M" (disentangling the continuous mask for Gaussian noise), (2) "-I_1 (repr. $I(X_S;S)$" (omitting the mutual information between data sample representations and the signal variable), (3) "-I_2 (repr. $I(X;X_S)$" (jettisoning the mutual information between data samples and their representations), (4) "-L_{VIB}" (excluding the pivotal Information Bottleneck mechanism that steers our methodology) and (5) "R-Bs" (replacing the bio-signals (BS) with random noise). Through a meticulous comparative analysis against the original *model*, we meticulously gauge the significance of each component in bolstering performance.

Each Component Makes Contributions to Performance: The discerning insights gleaned from Figure 4 unveil a compelling narrative: every facet of our methodology plays a pivotal role in shaping its performance landscape, thereby underscoring the indispensability of our holistic approach. Furthermore, it emerges distinctly that the Information Bottleneck (IB) mechanism (-L_{VIB}) emerges as the linchpin, exerting the most pronounced influence on the efficacy and potency of our methodology.

The Significance of Bio-signals: The significance of bio-signals (BS) in achieving optimal performance is evident when we observe a substantial decline in performance upon replacing them with random noise signals. This finding highlights the critical role of bio-signals in capturing the global, principle-based semantics inherent in biological datasets, like the dynamics of water pressure or dynamic cell areas. By incorporating bio-signals, our model gains a deeper understanding of the underlying biological processes, leading to more accurate representations.

Efficiency Comparison (RQ3)

This section compares the training time of different methods, including our model *BioDM*, and presents the results in Table 3. To be fair, all methods were implemented in

Python 3.9 using PyTorch 2.1 on a server with 48 cores and 8 Nvidia GeForce H800 GPUs. Our model *BioDM* demonstrates comparable efficiency compared to baseline methods across all three datasets (Wets, Tension and Fluid), while maintaining superior performance. This confirms that our model *BioDM* is well-suited for large-scale datasets and captures causal effects in biological datasets. Among the baseline models, SmoothGrad is the slowest due to its method of combining gradients from multiple inputs to create richer attribution maps, although it performs well. Thus, SmoothGrad is not suitable to be applied in real-life applications.

Impact and Limitations

R#CQAf-W3 This paper presents research that seeks to push the boundaries of artificial intelligence in scientific applications. Additionally, we make a significant contribution by providing three extensive scientific datasets that will greatly impact the development of AI in the field of science. Although our work carries potential societal implications, we recognize the importance of evaluating and addressing these consequences in a broader context that extends beyond the scope of this paper. Meanwhile, our method focus on theoretical proposals rather than practical implementation, potential challenges in scalability and generalizability of the proposed model to diverse biological datasets, and the need for further validation through real-world applications to demonstrate its effectiveness beyond theoretical frameworks.

Conclusion

Accurately capturing biological dynamics is crucial for understanding biological mechanisms, but existing generative models like DDPMs struggle due to limitations in capturing global semantics, disentangling causal effects, and the lack of large-scale datasets. Our proposed *BioDM*, a bio-signal-guided DDPM, addresses these limitations by leveraging the Information Bottleneck mechanism. *BioDM* effectively captures global semantics, identifies causal effects, and benefits from the large-scale BioDBench dataset. Extensive experiments demonstrate *BioDM*'s superior performance in capturing causal effects, advancing our understanding of biological dynamics.

References

Ahmadzadeh, A.; Kempton, D. J.; Chen, Y.; and Angryk, R. A. 2021. Multiscale iou: A metric for evaluation of salient object detection with fine structures. In *2021 IEEE International Conference on Image Processing (ICIP)*, 684–688. IEEE.

AI4Science, M. R.; and Quantum, M. A. 2023. The impact of large language models on scientific discovery: a preliminary study using gpt-4. *arXiv preprint arXiv:2311.07361*.

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*.

Ancona, M.; Ceolini, E.; Öztireli, C.; and Gross, M. 2017. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*.

Baehrens, D.; Schroeter, T.; Harmeling, S.; Kawanabe, M.; Hansen, K.; and Müller, K.-R. 2010. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11: 1803–1831.

Best, R. B.; and Hummer, G. 2011. Diffusion models of protein folding. *Physical Chemistry Chemical Physics*, 13(38): 16902–16911.

Cachay, S. R.; Zhao, B.; James, H.; and Yu, R. 2023. DYffusion: A Dynamics-informed Diffusion Model for Spatiotemporal Forecasting. *arXiv preprint arXiv:2306.01984*.

Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; and Abbeel, P. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29.

Chen, Y.; Yang, X.-H.; Wei, Z.; Heidari, A. A.; Zheng, N.; Li, Z.; Chen, H.; Hu, H.; Zhou, Q.; and Guan, Q. 2022. Generative adversarial networks in medical image augmentation: a review. *Computers in Biology and Medicine*, 144: 105382.

Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; and Bharath, A. A. 2018. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1): 53–65.

de Haan, C. A.; and Rottier, P. J. 2005. Molecular interactions in the assembly of coronaviruses. *Advances in virus research*, 64: 165–230.

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

Diogo, R. 2017. *Evolution driven by organismal behavior*. Springer.

Fu, C.; Yan, K.; Wang, L.; Au, W. Y.; McThrow, M.; Komikado, T.; Maruhashi, K.; Uchino, K.; Qian, X.; and Ji, S. 2023. A Latent Diffusion Model for Protein Structure Generation. *arXiv preprint arXiv:2305.04120*.

Gao, Z.; Tan, C.; and Li, S. Z. 2023. DiffSDS: A language diffusion model for protein backbone inpainting under geometric conditions and constraints. *arXiv preprint arXiv:2301.09642*.

Heumos, L.; Schaar, A. C.; Lance, C.; Litinetskaya, A.; Drost, F.; Zappia, L.; Lücken, M. D.; Strobl, D. C.; Henao, J.; Curion, F.; et al. 2023. Best practices for single-cell analysis across modalities. *Nature Reviews Genetics*, 24(8): 550–572.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Holzschuh, B.; Vegetti, S.; and Thuerey, N. 2023. Solving Inverse Physics Problems with Score Matching. In *ICML 2023 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*.

Kalfon, J.; Samaran, J.; Peyre, G.; and Cantini, L. 2024. scPRINT: pre-training on 50 million cells allows robust gene network predictions. *bioRxiv*, 2024–07.

Kang, L.; Gao, A.-K.; Han, F.; Cui, W.; and Lu, X.-Y. 2023. Propulsive performance and vortex dynamics of jellyfish-like propulsion with burst-and-coast strategy. *Physics of Fluids*, 35(9).

Kingma, D.; Salimans, T.; Poole, B.; and Ho, J. 2021. Variational diffusion models. *Advances in neural information processing systems*, 34: 21696–21707.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

MacDonald, J.; Wäldchen, S.; Hauch, S.; and Kutyniok, G. 2019. A rate-distortion framework for explaining neural network decisions. *arXiv preprint arXiv:1905.11092*.

Nishikawa, T.; Lee, M.; and Amau, M. 2024. New generative methods for single-cell transcriptome data in bulk RNA sequence deconvolution. *Scientific Reports*, 14(1): 4156.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4195–4205.

Peters, J.; Janzing, D.; and Schölkopf, B. 2017. *Elements of causal inference: foundations and learning algorithms*. The MIT Press.

Po, R.; Yifan, W.; Golyanik, V.; Aberman, K.; Barron, J. T.; Bermano, A.; Chan, E.; Dekel, T.; Holynski, A.; Kanazawa, A.; et al. 2024. State of the art on diffusion models for visual computing. In *Computer Graphics Forum*, volume 43, e15063. Wiley Online Library.

Price, I.; Sanchez-Gonzalez, A.; Alet, F.; Ewalds, T.; El-Kadi, A.; Stott, J.; Mohamed, S.; Battaglia, P.; Lam, R.; and Willson, M. 2023. GenCast: Diffusion-based ensemble forecasting for medium-range weather. *arXiv preprint arXiv:2312.15796*.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, 234–241. Springer.

Schulz, K.; Sixt, L.; Tombari, F.; and Landgraf, T. 2020. Restricting the flow: Information bottlenecks for attribution. *arXiv preprint arXiv:2001.00396*.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, S. P.; Wang, L.; Gupta, S.; Goli, H.; Padmanabhan, P.; and Gulyás, B. 2020. 3D deep learning on medical images: a review. *Sensors*, 20(18): 5097.

Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; and Wattenberg, M. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Southern, E.; Mir, K.; and Shchepinov, M. 1999. Molecular interactions on microarrays. *Nature genetics*, 21(1): 5–9.

Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

Tang, R.; Liu, L.; Pandey, A.; Jiang, Z.; Yang, G.; Kumar, K.; Stenetorp, P.; Lin, J.; and Ture, F. 2022. What the daam: Interpreting stable diffusion using cross attention. *arXiv preprint arXiv:2210.04885*.

Tishby, N.; and Zaslavsky, N. 2015. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, 1–5. IEEE.

Tishby Naftali, F. C., Pereira; and William, B. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

Van den Berge, K.; Roux de Bézieux, H.; Street, K.; Saelens, W.; Cannoodt, R.; Saeys, Y.; Dudoit, S.; and Clement, L. 2020. Trajectory-based differential expression analysis for single-cell sequencing data. *Nature communications*, 11(1): 1201.

Wang, S.-Y.; Efros, A. A.; Zhu, J.-Y.; and Zhang, R. 2023. Evaluating data attribution for text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7192–7203.

Watson, J. L.; Juergens, D.; Bennett, N. R.; Trippe, B. L.; Yim, J.; Eisenach, H. E.; Ahern, W.; Borst, A. J.; Ragotte, R. J.; Milles, L. F.; et al. 2023. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976): 1089–1100.

Wcislo, W. T. 2021. A dual role for behavior in evolution and shaping organismal selective environments. *Annual Review of Ecology, Evolution, and Systematics*, 52(1): 343–362.

Weymouth, G. D. 2015. Lily pad: Towards real-time interactive computational fluid dynamics. *arXiv preprint arXiv:1510.06886*.

Wu, K. E.; Yang, K. K.; Berg, R. v. d.; Zou, J. Y.; Lu, A. X.; and Amini, A. P. 2022. Protein structure generation via folding diffusion. *arXiv preprint arXiv:2209.15611*.

Wu, T.; Maruyama, T.; Wei, L.; Zhang, T.; Du, Y.; Iaccarino, G.; and Leskovec, J. 2024. Compositional Generative Inverse Design. In *The Twelfth International Conference on Learning Representations*.

Xu, M.; Yu, L.; Song, Y.; Shi, C.; Ermon, S.; and Tang, J. 2022. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*.

Yan, J. N.; Gu, J.; and Rush, A. M. 2024. Diffusion models without attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8239–8249.

Zhu, Y.; and Zhao, Y. 2023. Diffusion models in nlp: A survey. *arXiv preprint arXiv:2303.07576*.

Zou, H.; Kim, Z. M.; and Kang, D. 2023. A survey of diffusion models in natural language processing. *arXiv preprint arXiv:2305.14671*.

Reproducibility Checklist

This paper:

- Includes a conceptual outline and/or pseudocode description of AI methods introduced [Yes]
- Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results [Yes]
- Provides well marked pedagogical references for less-familiare readers to gain background necessary to replicate the paper [Yes]
- Does this paper make theoretical contributions? [Yes]
- All assumptions and restrictions are stated clearly and formally. [Yes]
- All novel claims are stated formally (e.g., in theorem statements). [Yes]
- Proofs of all novel claims are included. [Yes]
- Proof sketches or intuitions are given for complex and/or novel results. [Yes]
- Appropriate citations to theoretical tools used are given. [Yes]
- All theoretical claims are demonstrated empirically to hold. [Yes]
- All experimental code used to eliminate or disprove claims is included. ([Yes]
- Does this paper rely on one or more datasets? [Yes]

If yes, please complete the list below.

- A motivation is given for why the experiments are conducted on the selected datasets [Yes]
- All novel datasets introduced in this paper are included in a data appendix. [Yes]
- All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. [Yes]
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations. [Yes]

- All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available. [Yes]
- All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing. [Yes]
- Does this paper include computational experiments? [Yes]

If yes, please complete the list below.

- Any code required for pre-processing data is included in the appendix. [Yes]
- All source code required for conducting and analyzing the experiments is included in a code appendix. [Yes]
- All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. [Yes]
- All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from [Yes]
- If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. [Yes]
- This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. [Yes]
- This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. [Yes]
- This paper states the number of algorithm runs used to compute each reported result. [Yes]
- Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. [NA]
- The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). [Yes]
- This paper lists all final hyperparameters used for each model/algorithm in the paper's experiments. [Yes]
- This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting. [Yes]

Appendix

A.1 Derivation of empirical VIB

In this section, we provide the deduction progress for the following formula:

$$L_{\text{VIB}} = \mathbb{E}_{(X,S)\sim p(X,S)}\Big[\frac{1}{2}\big(1+\sum_{j=1}^{D}\big(\log\sigma^2_{X_S,\varphi}(X)- \mu^2_{X_S,\varphi}(X)-\sigma^2_{X_S,\varphi}(X)\big)_j\big)+\frac{1}{2}\beta||\mu_{S,\varphi}(X_S)-S||_2^2\Big] \tag{9}$$

Assuming that the prior $q_\varphi(X_S)$ and the posterior approximation $p_\varphi(X_S|X)$ are both Gaussian distributions, we proceed with the following notation. Let D represent the dimensionality of X_S. Through reparameterization trick, the variational mean and standard deviation evaluated on X are $\mu_{X_S,\varphi}(X)$ and $\sigma^2_{X_S,\varphi}(X)$, respectively. Consequently, we have the following relationship:

$$\begin{aligned}&\int q_\varphi(X_S)\log p(X_S)dX_S = \\ &\int \mathcal{N}(X_S;\mu_{X_S,\varphi}(X),\sigma^2_{X_S,\varphi}(X))\log\mathcal{N}(X_S;\mathbf{0},\mathbf{I})dX_S \\ &=-\frac{D}{2}\log(2\pi)-\frac{1}{2}\sum_{j=1}^{D}(\mu^2_{X_S,\varphi}(X)+\sigma^2_{X_S,\varphi}(X))_j\end{aligned} \tag{10}$$

And:

$$\begin{aligned}&\int q_\varphi(X_S)\log q_\varphi(X_S)dX_S = \\ &\int \mathcal{N}(X_S;\mu_{X_S,\varphi}(X),\sigma^2_{X_S,\varphi}(X))\log\mathcal{N}(X_S;\mu_{X_S,\varphi}(X), \sigma^2_{X_S,\varphi}(X))dX_S \\ &=-\frac{D}{2}\log(2\pi)-\frac{1}{2}\sum_{j=1}^{D}(1+\log\sigma^2_{X_S,\varphi}(X))_j\end{aligned} \tag{11}$$

Therefore, we can have the following deduction result:

$$\begin{aligned}&-D_{\text{KL}}(q_\varphi(X_S)||p_\varphi(X_S)) = \int q_\varphi(X_S)(\log p_\varphi(X_S)- q_\varphi(X_S))dX_S \\ &=\frac{1}{2}\big(1+\sum_{j=1}^{D}\big(\log\sigma^2_{X_S,\varphi}(X)-\mu^2_{X_S,\varphi}(X)-\sigma^2_{X_S,\varphi}(X)\big)_j\big)\end{aligned} \tag{12}$$

Thus, $\beta\log q_\varphi(S|X_S) = \beta\log(\mathcal{N}(S;\mu_{S,\varphi}(X_S),I)$ is deduced to a standard MSE loss, Thus, we can obtain the loss $L_{\text{VIB}} = \mathbb{E}_{(X,S)\sim p(X,S)}$ $\Big[\frac{1}{2}\big(1 + \sum_{j=1}^{D}\big(\log\sigma^2_{X_S,\varphi}(X) - \mu^2_{X_S,\varphi}(X) - \sigma^2_{X_S,\varphi}(X)\big)_j\big)+\frac{1}{2}\beta||\mu_{S,\varphi}(X_S)-S||_2^2\Big]$.

A.2 Fluid Data Generation

We employ the Lily-Pad simulator (Weymouth 2015) for generating both the training and testing datasets. The resolution of the 2D flow field is configured to be 128×128. It's noteworthy that in the context of Lily-Pad, the flow field is assumed to extend infinitely. The head of the jellyfish remains fixed at the coordinates $(25.6, 64)$. The representation of its two wings takes the form of identical ellipses, characterized by a fixed ratio of 0.15 between the shorter and longer axes. At every instant, symmetry is maintained across the central horizontal line defined by $y = 64$. To delineate the wing boundaries, we meticulously sample a total of $M = 20$ points along each wing. The pivotal parameter governing the jellyfish's control in this 2D experiment is the opening angle of the wings. This angle is defined as the deviation between the longer axis of the upper wing and the horizontal line. It serves as the crucial control signal, denoted as w.

Each trajectory originates with the widest possible opening angle and proceeds along a periodic cosine curve with a period of $T' = 200$. Trajectories are distinguished by variations in their initial angle, angle amplitude, and phase ratio, denoted as τ which represents the ratio between the closing duration and the entire pitching duration. For each trajectory, the initial angle w_0 is generated through a two-step process. Initially, a random mean angle $w^{(m)}$ is sampled within the range of $[20°, 40°]$. Subsequently, a random angle amplitude $w^{(a)}$ is sampled within the interval $[10°, \min(w^{(m)}, 60° - w^{(m)})]$. The resultant initial angle w_0 is then computed as $w_0 = w^{(m)} + w^{(a)}$, constrained within the range of $[10°, 60°]$. Meanwhile, the phase ratio τ is randomly chosen from the range of $[0.2, 0.8]$. The opening angle w_t at step t follows a specific pattern: it decreases from $w^{(m)} + w^{(a)}$ to $w^{(m)} - w^{(a)}$ as t advances from 0 to $\tau T'$, and then it increases from $w^{(m)} - w^{(a)}$ to $w^{(m)} + w^{(a)}$ as t progresses from $\tau T'$ to T'. Beyond this point, w_t exhibits periodic variations for $t > T'$. This configuration aligns with previous studies on the propulsive performance of jellyfish (Kang et al. 2023). Each trajectory is simulated for a total of 600 simulation steps, equivalent to 3 periods. To conserve space, only the segment of the trajectory spanning from $T' = 200$ to $3T' = 600$ steps is saved, with a step size of 10. This decision is made as the simulation from $t = 0$ to $T' = 200$ is primarily intended for initializing the flow field. Consequently, each trajectory is stored as a sequence comprising $\tilde{T} = (600 - 200)/10 = 40$ discrete steps.

In addition to tracking the positions of the wing boundary points and the opening angles w, we incorporate an image-like representation of the wing boundaries. This alternative representation contains valuable spatial information that can be more efficiently assimilated alongside the PDE states (fluid field) through convolutional neural networks. For each trajectory, this image-like boundary representation aligns seamlessly with the shape of the PDE states. At each time step, the boundaries of the two wings are combined and transformed into a tensor of dimensions [3, 64, 64]. Within each grid cell of this tensor, three distinct features are included: a binary mask indicating whether the cell

resides within a boundary (marked as 1) or within the fluid (denoted as 0), and a relative position $(\Delta x, \Delta y)$ representing the distance from the cell center to the nearest point on the boundary. This representation enhances the compatibility between boundary information and PDE states. For every trajectory, we retain data on PDE states, opening angles, boundary points, boundary masks, offsets, and force data. These components are specified as follows:

- PDE states u: These have a shape of $[\tilde{T}, 3, 64, 64]$, representing the fluid field states for each time step, including velocity in the x and y directions and pressure. To conserve space, we downsample the resolution from 128×128 to 64×64.
  - velocity: $[\tilde{T}, 2, 64, 64]$.
  - pressure: $[\tilde{T}, 1, 64, 64]$.
- opening angels w: they have a shape of $[\tilde{T}]$. For each step, we save the opening angle in radians.
- boundary points: shape $[\tilde{T}, 2, M, 2]$. With a shape of $[\tilde{T}, 2, M, 2]$, we record the boundary points for both the upper and lower wings. Each wing comprises $M = 20$ points, and each point has 2 coordinates. To ensure compatibility with the downsampling of states, the coordinates in the x and y directions are scaled down to half $(64/128)$ of their original values.
- boundary mask and offsets b: They have a shape of $[\tilde{T}, 3, 64, 64]$. For each time step, this includes a mask indicating the merged wings along with the half coordinates of boundary points and offsets in both the x and y directions. The resolution is 64×64, matching that of the PDE states.
  - mask: $[\tilde{T}, 1, 64, 64]$.
  - offsets: $[\tilde{T}, 2, 64, 64]$.
- force: it has a shape of $[\tilde{T}, 2]$. For each step, the simulator outputs the horizontal and vertical force from fluid to the jellyfish. The horizontal force is regarded as a thrust to jellyfish if positive and a drag otherwise.

We generated a total of $n = 45,000$ trajectories, each distinguished by varying parameters such as $w^{(a)}, w^{(m)}$, and τ. Each trajectory occupies approximately 2MB of storage space, contributing to an overall dataset size of around 100GB. To create training samples, we employed sliding time windows that encompassed $T = 20$ consecutive time steps of both states and boundaries. This configuration corresponds to $T^{'} = 200$ original simulation steps, constituting a complete wing movement period. Consequently, each trajectory could generate up to 20 individual samples, resulting in a grand total of 6 million training samples. In each training sample, the opening angle remained consistent between the initial and final time steps due to the periodic nature of the motion. This consistency served as the control condition for our experiments. For test trajectories, we carefully selected the opening angle of the jellyfish at the initial time and used the initial states as the control conditions for both the initial and final time and state configurations.

A.3 Related Work

Denosing Diffusion Probabilistic Model (DDPM) for AI for Science. Denoising diffusion probabilistic models have demonstrated their capability to predict the dynamic evolution in various domains such as fluid dynamics (Cachay et al. 2023), weather forecasting (Price et al. 2023), and molecular dynamics (Wu et al. 2022). They have also been effectively applied in inverse design tasks, enabling the optimization of airfoils (Wu et al. 2024) and proteins (Watson et al. 2023). Additionally, diffusion models have shown promise in solving complex inverse problems (Holzschuh, Vegetti, and Thuerey 2023). These are just a few examples of the diverse applications where diffusion models have been successfully employed. In the realm of biology, researchers have utilized the DDPM to model diffusion processes in biological networks (Fu et al. 2023; Best and Hummer 2011; Gao, Tan, and Li 2023; Xu et al. 2022), enabling the analysis of protein-protein interactions and gene regulatory networks. In the field of physics, the DDPM has been applied to study the diffusion of particles in complex systems, such as the spread of heat in materials. Furthermore, in the domain of chemistry, the DDPM has been employed to understand the diffusion of molecules and reactions in chemical systems. These studies highlight the versatility and effectiveness of the DDPM in capturing and analyzing diffusion dynamics across various scientific disciplines (Xu et al. 2022). Ongoing research aims to further explore its potential for solving complex problems in AI for Science.

Diffusion Probabilistic Model. Diffusion Probabilistic Models (DM) (Ho, Jain, and Abbeel 2020) have emerged as the leading approach in density estimation (Kingma et al. 2021) and have also demonstrated superior sample quality (Dhariwal and Nichol 2021). These models leverage the inherent characteristics of image-like data by employing a UNet as their underlying neural backbone (Ronneberger, Fischer, and Brox 2015; Ho, Jain, and Abbeel 2020; Dhariwal and Nichol 2021). Notably, the use of a reweighted objective (Ho, Jain, and Abbeel 2020) during training typically leads to the highest synthesis quality. Another research line for image generation is GAN-based methods (Creswell et al. 2018; Chen et al. 2016). A representative study of this research line is infoGAN (Chen et al. 2016), which is a type of generative adversarial network that not only generates realistic samples but also maximizes the mutual information between a select few latent variables and the generated output. InfoGAN allows for the discovery of meaningful representations. However, these studies cannot provide explanations for the generated samples.

A.4 Preliminary on Denoise Diffusion Probabilistic Model

The Denoising Diffusion Probabilistic Model (DDPM) (Ho, Jain, and Abbeel 2020) consists of two essential processes: the forward process (or diffusion process) and the reverse process. Let's focus on describing the forward process first. The forward process in a diffusion model approximates the posterior distribution $q(x_{1:T}|x_0)$, which represents the sequence of latent variables $x_{1:T}$ given an initial value x_0. This

approximation is achieved by iteratively applying a Markov chain that adds Gaussian noise gradually over time. Specifically, the forward process is represented as follows:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \mu_t(x_{t-1}), \beta_t \mathbf{I}\right) \tag{13}$$

Here, x_t denotes the latent variable at time step t, and x_{t-1} is the variable at the previous time step. The distribution $q(x_t|x_{t-1})$ is modeled as a Gaussian distribution with mean $\mu_t(x_{t-1})$ and variance $\beta_t \mathbf{I}$, where β_t is the variance parameter at time step t, and $\mathbf{I}$ represents the identity matrix. The mean $\mu_t(x_{t-1})$ can depend on the previous latent variable x_{t-1} and is typically modeled using neural networks or other parameterized functions. By sequentially applying the distribution $q(x_t|x_{t-1})$ for each time step, starting from the initial value x_0, we obtain an approximation of the posterior distribution $q(x_{1:T}|x_0)$ that captures the temporal evolution of the latent variables via $q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1})$.

To describe the reverse process, let's consider a diffusion model with T time steps. Given an observed data point x_T at the final time step, the goal is to generate a sample from the initial distribution $p(x_0)$. The reverse process in a diffusion model can be formulated as follows: (1) Initialization: Set x_T as the observed data point. (2) Iterative Sampling: Starting from $t = T-1$ and moving backwards until $t = 0$, sample x_t from the distribution $p(x_t|x_{t+1})$, where $p(x_t|x_{t+1})$ represents the reverse diffusion process.

The distribution $p(x_t|x_{t+1})$ in the reverse process is typically modeled as a Gaussian distribution, similar to the forward process. However, the mean and variance parameters are adjusted to account for the reverse direction. The specific form of $p(x_t|x_{t+1})$ is defined as follows:

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{14}$$

$$p(x_t|x_{t+1}) := \mathcal{N}(x_t; \mu_\theta(x_{t+1}, t+1), \sum_\theta(x_{t+1}, t+1))$$

By iteratively sampling from the reverse process, we can generate a sequence of latent variables $x_{0:T}$ that follows the reverse diffusion process. This reverse sequence represents a sample from the initial distribution $p(x_0)$. The reverse process is crucial for training the diffusion model. During training, the model learns to approximate the reverse process by minimizing the discrepancy between the generated samples and the observed data points. This training procedure ensures that the model captures the underlying data distribution and can generate realistic samples.

The optimization objective of the diffusion model is conducted via the following negative log likelihood:

$$\begin{aligned} \mathbb{E}[-\log p_\theta(x_0)] &\leq \mathbb{E}_q[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}] \\ &= \mathbb{E}_q[-\log p(x_T) - \sum_{t\geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}] =: \mathcal{L} \end{aligned} \tag{15}$$

A.5 Evaluation Metrics

Building upon previous research (Schulz et al. 2020; Ancona et al. 2017; Ahmadzadeh et al. 2021), we have adopted Intersection over Union (IoU) and Sensitivity-n as the evaluation metrics for all the methods in our study. These metrics are widely used in the field and provide valuable insights into the performance of the approaches. The definition of IoU is shown as follows:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{16}$$

IoU is a value between 0 and 1, where a higher value indicates a better overlap between the predicted and ground truth regions. The rationale behind utilizing IoU is to calculate the intersection over union between the generated images and the ground-truth cases, considering the corresponding signal variances such as cell areas or circumferences.

The Sensitivity-n metric, introduced by Ancona et al. (2017), evaluates attribution methods by randomly masking network inputs and quantifying the correlation between the masked attribution and the corresponding decrease in classifier score. For regression task, Sensitivity-n computes the Pearson correlation coefficient between the attribution values and the associated deviation in prediction when masking a set M_n of n randomly selected pixel indices:

$$\text{Sensitivity-n} = \text{corr}\left(\sum_{i\in M_n} R_i(x), \left(S(x) - S(x_{[x_{M_n}=0]})\right)^2\right) \tag{17}$$

Here, R_i represents the relevance of the indexed pixel i, $S(x)$ denotes the prediction from the decode network with clean inputs x, and $S(x_{[x_{M_n}=0]})$ represents the prediction value from the decode network after randomly setting the masked pixels to 0. Intuitively, by masking random pixels, the prediction $S(x_{[x_{M_n}=0]})$ will deviate from the prediction $S(x)$ that is obtained without masking the pixels. Sensitivity-n quantifies how the reduction of the prediction performance $\left(S(x) - S(x_{[x_{M_n}=0]})\right)^2$ is correlated with the total relevance $\sum_{i\in M_n} R_i(x)$ a method assigns to the masked pixels. A good attribution method should assign high relevance values to the pixels such that when those pixels are masked, the prediction performance drop significantly, producing a high Sensitivity-n value.

A.6 Details of baselines' implementations

To help reproduce the experiment results of all methods, we also provide detailed implementations of all baselines in Appendix . With the assistance of the mask M obtained through the regression function $\mu_{S,\varphi}$, the methods of our baselines focus on denoising only the features that are considered relevant according to the mask M.

$$L_{\epsilon_\theta, M} = ||\epsilon \odot M - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||_2^2, \epsilon \sim \mathcal{N}(0, I) \tag{18}$$

$$L_{\text{Reg}} = \mathbb{E}_{(X,S)\sim p(X,S)}\left[||\mu_{S,\varphi}(X_S) - S||_2^2\right]$$

Combined together

$$L_{\text{baseline}} = L_{\epsilon_\theta, M} + \frac{1}{2} L_{\text{Reg}} \tag{19}$$

We employed seven distinct methods for obtaining masks:

- Random: Random involves utilizing random numbers as masks.
- Gradient: The gradient of x_t is employed as the input to the model, and the gradient of x_t is obtained through backpropagation, with the model adopting the architecture of Unet U_φ.

$$M = \frac{\partial U_\varphi(x_t)}{\partial x_t} \tag{20}$$

- Guided Backpropagation: Guided Backpropagation is a technique used in interpretability of neural networks. It enhances the visualization of feature importance by guiding the backpropagation process through the positive gradient values and setting negative gradient values to zero. The formulation for Guided Backpropagation is as follows:

$$\frac{\partial U_\varphi(x_t)}{\partial x_t} > 0 \Rightarrow M^+(x_t) = \frac{\partial U_\varphi(x_t)}{\partial x_t} \tag{21}$$

For negative gradients:

$$\frac{\partial U_\varphi(x_t)}{\partial x_t} < 0 \Rightarrow M^-(x_t) = 0 \tag{22}$$

- SmoothGrad is a technique used to reduce noise in the interpretation of deep neural network predictions by averaging gradients over multiple perturbed instances of the input. The formulation for SmoothGrad is as follows:

$$M = \frac{1}{N}\sum_{j=1}^{N} \frac{\partial U_\varphi(\text{Perturb}(x_t))}{\partial x_{i,t}} \tag{23}$$

$Perturb(x)$ is a function that introduces random noise or perturbation to the input x.

- GuidedCAM: Guided Class Activation Mapping (GuidedCAM) is a technique used for visualizing and interpreting the decisions of a convolutional neural network (CNN). It is often applied to understand the importance of different regions in an input image for a particular class prediction. The formulation for GuidedCAM is as follows:

$$M = ReLU(\frac{\partial U_\varphi(x_t)}{\partial x_t} \odot U_\varphi(x_t)) \tag{24}$$

- Guided Grad-CAM (Guided Gradient-weighted Class Activation Mapping) combines the concepts of Guided Backpropagation and Grad-CAM to highlight important regions in an input image for a specific class prediction. The formulation for GuidedGradCAM is as follows:

$$M = ReLU(\frac{\partial U_\varphi(x_t)}{\partial x_t} \odot U_\varphi(x_t)) \odot ReLU(\frac{\partial U_\theta(x_t)}{\partial x_t} \odot U_\theta(x_t)) \tag{25}$$

- Saliency is an attribution method used to understand the importance of input features for a given model prediction. The formulation for Integrated Gradients is as follows:

$$M = (x_t - x_t^{'}) \odot \int_{\alpha=0}^{1} \frac{\partial U_\varphi(Interpolate(x^{'}, x, \alpha)}{\partial x_t} d\alpha \tag{26}$$

Table 4: The Sensitivity-n values of all methods were assessed on two datasets, specifically tension and wetting. The range of masked pixels varied from 10% to 90%. The definition of Sensitivity-n can be found in Equation 10. It was observed that the Sensitivity-n values did not exhibit significant differentiation among the baseline methods, as depicted in Figure 3. We have averaged the results shown in Figure 3 in this table. However, our proposed method *BioDM* consistently outperformed the others for most masking percentages. This demonstrates that our method can effectively identify relevant pixels that have a strong correlation with the signal's prediction.

Datasets	Wetting		Tension		Fluid
-	Area	Circumference	Area	Circumference	Pressure
Method	Sensitivity-n	Sensitivity-n	Sensitivity-n	Sensitivity-n	Sensitivity-n
Random	0.0423	0.1326	0.0541	0.0450	0.1992
Gradient	0.0200	0.0269	0.0024	0.0154	0.0107
Saliency	0.0315	0.0927	0.0189	0.0286	0.1653
GuidedBP	0.0627	0.1224	0.0343	0.0158	0.1783
GuidedCAM	0.0203	0.1375	0.0230	0.0081	0.1804
SmoothGrad	0.0279	0.1003	0.0204	0.0436	0.0132
GuidedGrad	0.0687	0.1057	0.0172	0.0482	0.1953
GuidedGrad	0.0687	0.1057	0.0172	0.0482	0.1953
Custom Diffusion	0.0361	0.0235	0.0283	0.0210	0.0398
DAAM	0.0210	0.0244	0.0204	0.0119	0.0307
***BioDM* (ours)**	**0.2857**	**0.1568**	**0.1201**	**0.0610**	**0.3181**

$Interpolate(x^{'}, x, \alpha)$ is a function that linearly interpolates between the baseline $x_t^{'}$ and the actual input x_t at a given scale α. The baseline $x_t^{'}$ is the mean of x_t.

Considering all of the aforementioned baselines, we have constrained the range of M to be clamped within the interval $[0.7, 1.0]$. This adjustment aids the baselines in achieving improved performance across all datasets.

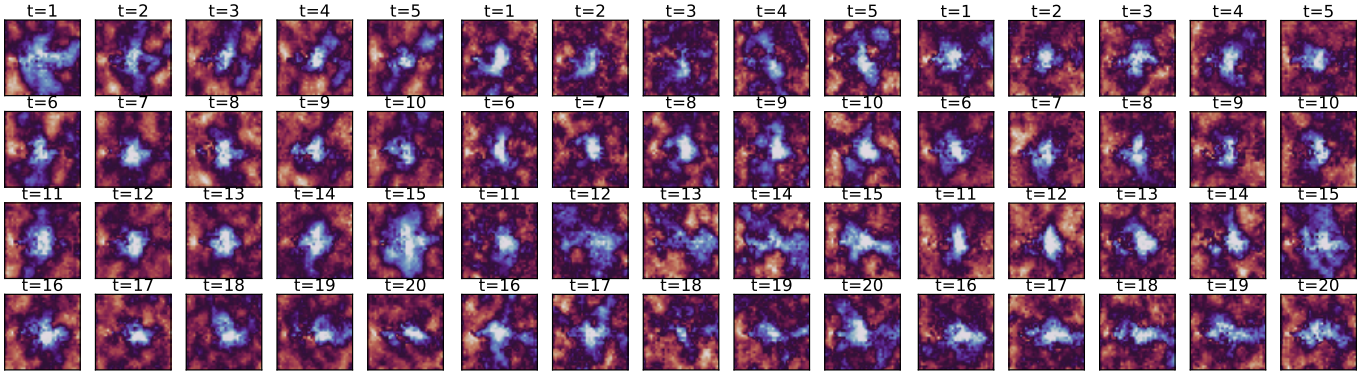
A.7 Training Details and Parameter Settings of BioDM

To ensure fairness and consistency in our experiments, we made several design choices regarding the configuration of our model. The hidden dimension of the U-Net neural network was set to 64, providing an appropriate level of complexity for the task at hand. To balance the contribution of the Information Bottleneck loss (L_{VIB}), we assigned a weight λ of 0.1 in Eq. 6. Additionally, we set the value of β to 1 for all datasets, promoting a suitable trade-off between reconstruction accuracy and information preservation. For the Gaussian diffusion process, we performed 1000 diffusion steps to allow for comprehensive information exchange. The channel size multiplier of the U-Net neural networks was set to $[1, 2, 4, 8]$, ensuring effective feature extraction across various scales. The number of channels for the two cell dynamics datasets and the fluid dataset was set to 20 and 42, respectively, accommodating the unique characteristics of each dataset. Three datasets were standardized to an image size of 32×32 pixels. To facilitate convergence, we adopted a learning rate of 8×10^{-5}.

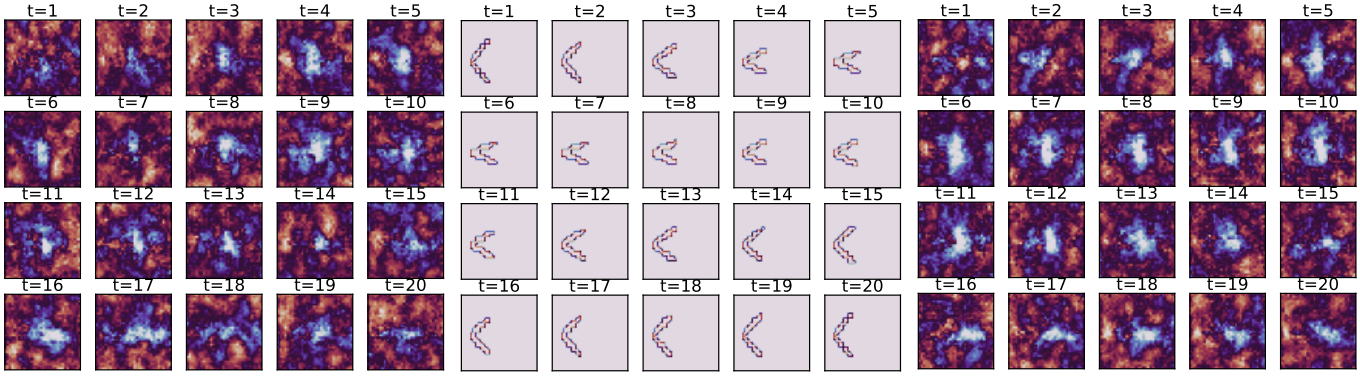
A.8 About causalityBioDM

Here we introduce the following proposition, which states that under certain conditions, the learned variable W is causal to the bio-signal S.

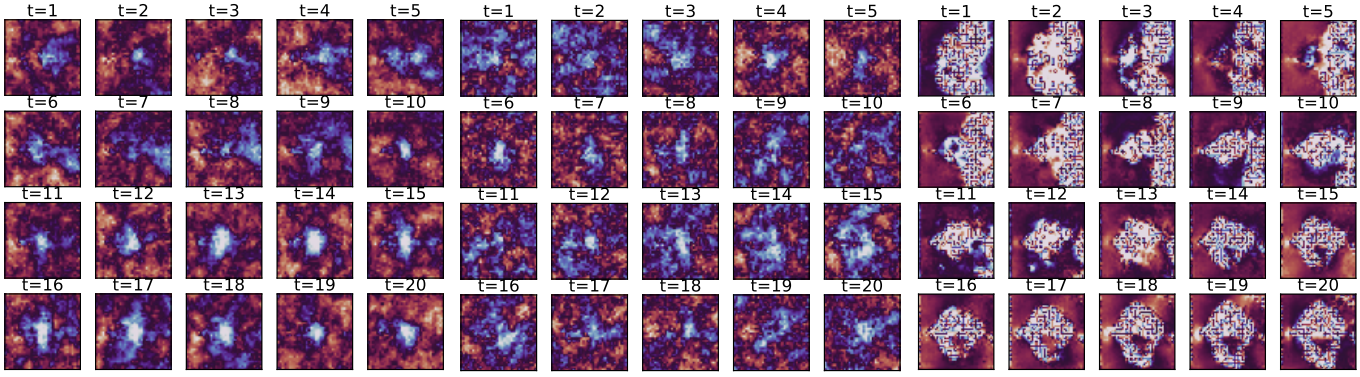
R#J6Qk-W1, R#Ai1S-Q1, R#4aBP-Q1, R#xZiv-W2



(a) Random (b) Gradient (c) Saliency

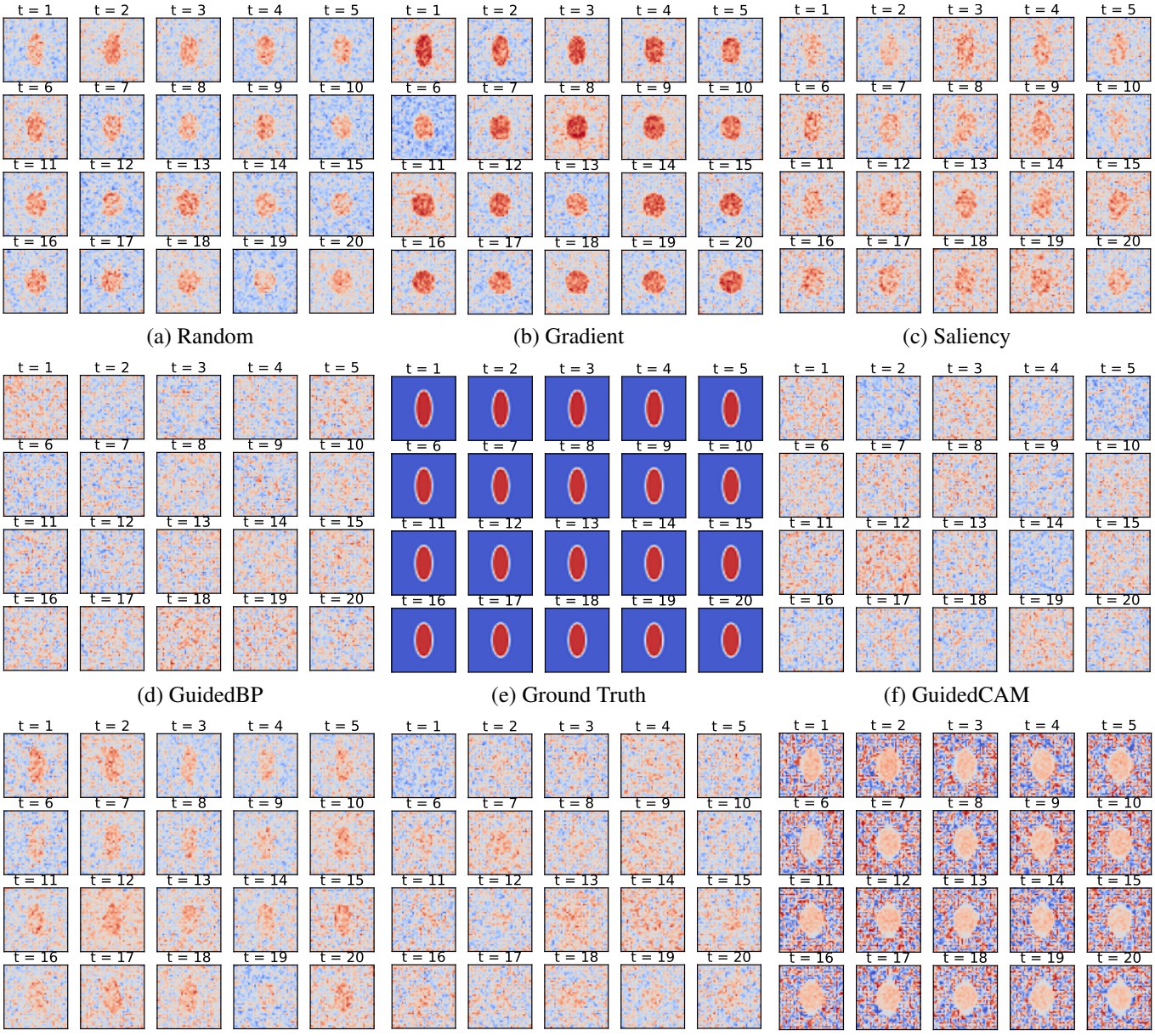


(d) GuidedBP (e) Ground Truth (f) GuidedCAM



(g) SmoothGrad (h) GuidedGrad (i) ***BioDM* (ours)**

Figure 6: We present visualizations comparing our proposed method, *BioDM*, with all baselines, on the Fluid dataset. The central subfigure represents the actual boundary of a jelly-like robot, which serves as a condition for generating the fluid pressure fields, denoted as X. The other subfigures depict the generated pressure field X by our *BioDM* and other baselines. The blended colors indicate the irrelevant parts of the pressure fields to the force measurement signal S, as identified by the methods, while the smoother colors represent the relevant parts. We observe that our *BioDM* accurately identifies that the enclosing region of the pressure fields is most relevant to the force measurement on the boundary. Furthermore, the generated pressure field exhibits consistent angles with respect to the given boundary. In contrast, the other baselines fail to generate a consistent pressure field and struggle to identify the relevant pixels associated with the signal.

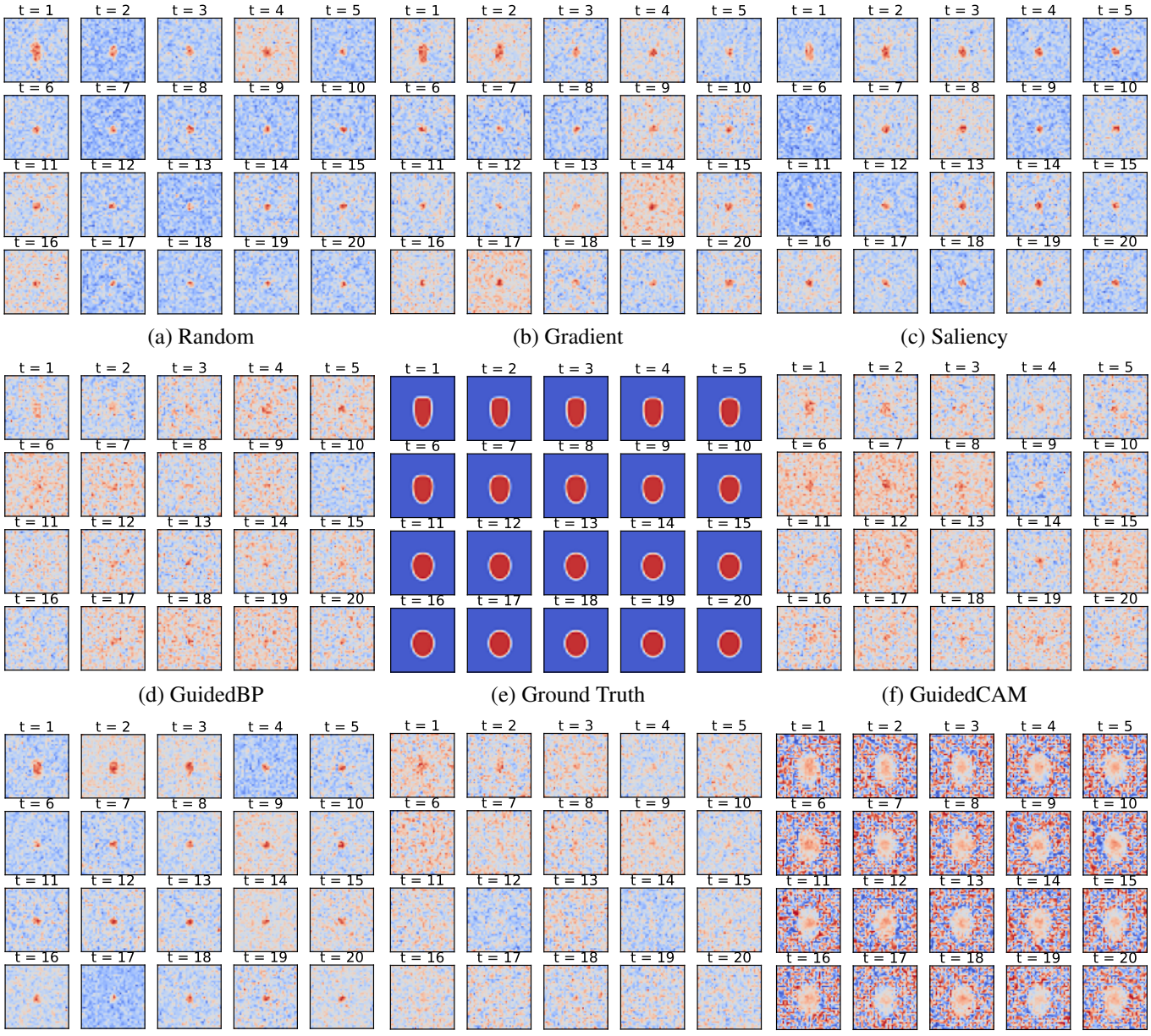


(a) Random (b) Gradient (c) Saliency

(d) GuidedBP (e) Ground Truth (f) GuidedCAM

(g) SmoothGrad (h) GuidedGrad (i) ***BioDM* (ours)**

Figure 7: We showcase visualization results that compare our proposed method, *BioDM*, with all baselines on the Tension dataset, specifically focusing on cells' area. The central portion represents the ground truth scenario, while the other figures display the generated results of 20 time steps using *BioDM* and all baselines, respectively. The mixed colors indicate the irrelevant portion to the signal variable (cells' area), while the smoother and more uniform color represents the relevant parts. Methods that generate cells more similar to the ground truth demonstrate higher explanatory capabilities. In this regard, *BioDM* exhibits superior explanatory ability as the generated images at each time step closely resemble those of the ground truth.



(a) Random (b) Gradient (c) Saliency

(d) GuidedBP (e) Ground Truth (f) GuidedCAM

(g) SmoothGrad (h) GuidedGrad (i) ***BioDM* (ours)**

Figure 8: We present visualizations that compare the performance of our proposed method, *BioDM*, with various baselines on the Wetting dataset. Specifically, we focus on the analysis of cells' circumference. The central portion of the visualizations represents the ground truth scenario, while the remaining figures depict the generated results from *BioDM* and the baselines at different time steps. The use of mixed colors in the visualizations helps distinguish the irrelevant portions of the signal variable (cells' circumference), while the smoother and more uniform colors highlight the relevant parts. Methods that generate cell structures closely resembling the ground truth demonstrate higher explanatory capabilities. In this context, our *BioDM* exhibits superior explanatory ability as the generated images at each time step closely resemble those of the ground truth.

Proposition 1. Assume that biological dynamics X contains a subset of its features that is causal to the bio-signal S. Let W be the features of X such that[5] $I(X;W) = I(X;S)$ and $I(S;X\backslash W) = 0$. Then we have W is causal to S and $X\backslash W$ is non-causal to S.

Proof. Here we use the causality as defined by Pearl (Peters, Janzing, and Schölkopf 2017). Concretely, X is a direct causal of S if there exist two values x and x' of X such that after the intervention, the corresponding S is different, while other variables are kept the same. Let the features of X be $\{x^{(1)}, x^{(2)}, ...x^{(n)}\}$. Denote indices of the features for W as I_W, and the indices of the features for $X\backslash W$ as $I_{X\backslash W}$. We have $I_W \cup I_{X\backslash W} = \{1, 2, ...n\}$ and $I_W \cap I_{X\backslash W} = \emptyset$. Since $I(S;X\backslash W) = 0$, we have that $\forall i \in I_{X\backslash W}, x^{(i)} \perp\!\!\!\perp S$ (independence). Therefore, for any $i \in I_{X\backslash W}$, the intervention of $x^{(i)}$ will not influence S, thus $X\backslash W$ is non-causal to S.

Also, by the assumption that X is causal to S, and the definition of Pearl causality, we know that there exists two values x and x' of X such that after the intervention, the corresponding S is different. Let I_{diff} be the indices of x and x' such that the corresponding features are different. We have that $I_{\text{diff}} \cap I_W \neq \emptyset$. Otherwise $I_{\text{diff}} \cap I_W = \emptyset$ will lead to $I_{\text{diff}} \subseteq I_{X\backslash W}$ (meaning that the difference of x and x' are all in features of $X\backslash W$, and contradict the assumption that $I(S;X\backslash W) = 0$. Therefore $I_{\text{diff}} \cap I_W \neq \emptyset$. Therefore, there exists two values w and w' of W such that after the intervention, the corresponding S is different, i.e., W is causal to S.

[5]Here $X\backslash W$ means X excluding W.