TEMPORAL ENVIRONMENT-AWARE IMAGE GENERATION VIA LATENT DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-cost cameras have recently become widely used to monitor environmental ecosystems. This paper focuses on scene prediction for monitoring small streams, which is critical for ensuring water supply and informing early actions for floods and droughts. In contrast to traditional stream models that typically rely on coarse-resolution weather data, stream images provide detailed information about water properties and local environment at a higher temporal frequency. This paper presents a multi-modal generative framework designed for frequent temporal stream imagery datasets, aimed at generating the subsequent stream images. This task is challenging due to the variability of stream images caused by changes in time and local environmental conditions. Our method captures scene changes in both stream and surrounding environment by incorporating temporal context of weather, water flow, and time information. We also introduce a domain-discriminative learning approach to enforce the learning of domain-specific information in generating images. Our experiments demonstrate the superior performance of the proposed method in preserving semantics of water and environmental properties, using real data from the West Brook area in western Massachusetts, USA.

1 INTRODUCTION

Effective observation over environmental ecosystems is critical for the sustainability of our planet, especially given the increasing pressures from environmental degradation, climate change, population growth, and urbanization. Remote sensing observations have been widely used to monitor environmental changes at large scale, but they are not suitable for monitoring small local regions due to limited spatial resolution and occlusions of various sources. More recently, low-cost cameras have been deployed in many environmental applications to consistently monitor target small regions at a high temporal frequency. These data sources provide rich information to facilitate the understanding of the underlying processes and timely decision making on managing natural resources.

This paper focuses on monitoring small headwater streams at a sub-hourly scale. Effective monitoring of these streams are critical for the estimation of multiple water quantity and water quality variables (e.g., streamflow, water depth, algal blooms), which are needed for ensuring drinking water supply for large population and suitable habitats for aquatic life. Because of the importance of this problem, scientists have built many different machine learning (ML)-based stream models (Feng et al., 2020; Cigizoglu, 2005; Jia et al., 2021; Karpatne et al., 2017). However, existing models highly rely on meteorological data and remote sensing data as input features, but such data sources are only available at a coarse resolution that is insufficient for predicting small streams. Besides, these data sources do not contain the information of local catchment conditions, which are critical for understanding the variability of water dynamics across different stream sites.

To enhance small stream monitoring, scientists have started using low-cost cameras to capture images at frequent intervals, such as sub-hourly, at ecologically important stream locations (Cam; Geo). In this paper, we introduce a new dataset containing stream images that are frequently collected from multiple sites in the West Brook area, western Massachusetts, USA. This dataset provides new opportunities for building ML models to extract detailed water-related information and the surrounding environmental conditions. Moreover, we propose Environment-Aware Latent Diffusion Model (EALDM), which is a new scene-predicting method to to generate subsequent stream images from previous stream images. This method can facilitate the forecasting of water and environmental conditions in future time, the imputation and restoration of noisy or blocked images in history for retrospective analysis, as well as many downstream prediction tasks for water properties (e.g., streamflow). However, this generative task is challenging as it requires the model to effectively capture not only varying backgrounds and views in different sites, but also the strong temporal variation in images due to changes in time and environmental conditions.

Here we summarize the contributions made by the proposed EALDM in addressing these challenges. In particular, EALDM extends the latent diffusion model by incorporating the temporal information and introduces three key in-

novations. First, EALDM integrates weather data, measured flow data, and camera images, which are from different sources and of different modalities, into the image generative process. The goal is to ensure that the changes observed in the generated images relative to the previous images are consistent with local weather conditions and water properties. Second, the image generative model explicitly takes into account time features, such as time of day and day of year, to learn from sub-hourly images taken over long periods. Once trained, EALDM can be used to generate images at specific times and capture seasonal and daily variations. Finally, we introduce a domain-discriminative learning method, which enforces the learning of background differences across multiple stream sites and improves the image generation towards a specific site. By leveraging temporal and environmental data, the model goes beyond mere image editing by predicting future changes in the environment and generating corresponding images, effectively providing a forward-looking view of the scene.

Our implementation can be accessed through the link `https://github.com/Anonymous/EALDM`. Our evaluations in small headwater streams in western Massachusetts demonstrate that EALDM successfully generates site-specific images at target times, which precisely capture water flows and environmental changes. This innovative approach has potential to significantly enhance our ability to monitor and predict streamflow, contributing to improved water management and ecological studies.

2 Related work

Stream modeling enables the prediction of water quantity and water quality variables, and thus can facilitate a range of decision making processes in managing water resources. With the success of ML over the past two decades, there is a growing interest in using data-driven ML techniques for water dynamics prediction (Feng et al., 2020; Cigizoglu, 2005; Jia et al., 2021; Karpatne et al., 2017; Ghosh et al., 2022). These methods typically use weather drivers (e.g., precipitation, air temperature, solar radiation) as input features, which are often available at a coarse resolution and insufficient for predicting small streams. Image-based streamflow prediction methods emerge as promising alternatives due to their ability to leverage visual data directly. The paper (Zhao et al., 2024) introduces a model combining Convolutional Neural Networks (CNNs) with Recurrent Neural Networks (RNNs) networks to capture both spatial and temporal dynamics in sequential stream image. This approach allows CNNs to extract spatial features from images while RNNs model the temporal dependencies, enhancing the accuracy of streamflow prediction.

Scene prediction for environmental monitoring aims to provide more comprehensive and detailed information by leveraging the temporal and spatial characteristics of environmental data. Diffusion models (Nichol et al., 2021; Ramesh et al., 2022; Saharia et al., 2022; Song et al., 2020; Dhariwal & Nichol, 2021; Jia et al., 2023) are a class of generative models that gradually transform a simple distribution (usually Gaussian noise) into a complex data distribution (such as images) through a process that iteratively denoises the data. This process is modeled as a Markov chain and involves hundreds to thousands of steps, each slightly denoising the input towards the data distribution.

Latent diffusion model (LDM) (Rombach et al., 2022) enhances the efficiency of traditional diffusion models by operating in a latent space instead of the pixel space. This is achieved by first encoding the data (images) into a lower-dimensional latent representation using an encoder (part of an autoencoder). The diffusion process is then applied in this latent space, and finally, a decoder transforms the denoised latent representation back into the image space. This approach significantly reduces the computational cost and can speed up the generation process without sacrificing the quality of the generated images. Classifier-Free Diffusion Guidance Model (CFDG) (Ho & Salimans, 2022) extends LDMs by enabling conditional generation without relying on an external classifier. Instead, the model conditions itself on given information (like text, class labels, etc.) during the training process, making it more efficient and easier to use for conditional generation tasks.

Recent advancements in controlled image diffusion models have enhanced the personalization and precision of image generation. Various techniques have emerged, offering diverse methods to manipulate these models for specific outputs. For instance, DreamBooth (Ruiz et al., 2023) fine-tunes the diffusion model based on small, user-specific datasets to personalize content. Text-guided methods modify prompts, CLIP features, and cross-attention (Brooks et al., 2023; Hertz et al., 2022; Kawar et al., 2023), enabling more detailed and context-sensitive image generation. Controlling the diffusion process allows adjustments over features such as inpainting (Avrahami et al., 2023) and color variation (Meng et al., 2021). Other techniques, such as GLIGEN (Li et al., 2023b), train attention layers to better guide grounded image creation.

ControlNet (Zhang et al., 2023) further extends the potential for conditional control in large pre-trained models, such as Stable Diffusion, by freezing the original model's parameters and training a separate copy to learn conditional controls without sacrificing image quality. This copy is used for learning diverse conditional controls, with the two networks connected via zero convolution layers whose weights are initialized to zero, allowing them to gradually

adjust during training. Similarly, BLIP-Diffusion (Li et al., 2024) introduces subject-driven generation, integrating a vision-language encoder (BLIP-2 (Li et al., 2023a)) with Stable Diffusion to guide the model using both image and text inputs, improving subject-driven generation and editing capabilities.

3 PROBLEM STATEMENT AND PRELIMINARIES

The objective of EALDM is to generate the image at the next time step $n+1$ conditioned on the available images, current weather information, and water flow data. This is formulated as an image generation problem where the goal is to accurately predict future scenes that incorporate critical environmental variables. Specifically, at each time step n, we represent the current image as $\mathbf{x}^n$, the weather data as $\mathbf{w}^n$, and the streamflow measurement as y^n. The objective is to learn a model f capable of generating the subsequent image f: $\{\mathbf{x}^n, \mathbf{w}^n, y^n\} \rightarrow \mathbf{x}^{n+1}$. These data sources are also collected over different stream sites (i.e., domains). We train the diffusion model using the collection of samples from different sites and omit the domain index when it does not cause ambiguity. We will use the subscript i to represent each stream site when we introduce the domain-discriminative learning method (Section 4.2).

Our proposed EALDM method is based on the latent diffusion model (LDM), which utilizes an encode-decoder architecture to covert images to the latent space and recover the images, while involving a UNet (Ronneberger et al., 2015)-based conditional diffusion model for conditional generation in the latent space. First, a given image $\mathbf{x}_0$ is encoded into a latent representation $\mathbf{z}_0 = E(\mathbf{x}_0)$ using the encoder E employed in latent diffusion model. Noise is progressively added to the latent representation through a forward process, and the obtained noisy data at t^{th} diffusion step model can be represented as $q(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t; \sqrt{\bar{\alpha}_t}\mathbf{z}_0, (1-\bar{\alpha}_t)\mathbf{I})$, where $\bar{\alpha}_t$ controls the noise schedule. The reverse process gradually recovers the latent representation by following an iterative denoising process $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_{t-1}; \mu_\theta(\mathbf{z}_t, t), \Sigma_\theta(\mathbf{z}_t, t))$, for $t = T$ to 1. The training process aims to minimize the loss: $\mathcal{L} = \mathbb{E}_{t,\mathbf{z}_0,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)\|^2\right]$, where $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t)$ is the predicted noise, and $\boldsymbol{\epsilon}$ is the actual noise. After denoising, $\mathbf{z}_0$ is decoded back to the image $\mathbf{x}_0 = D(\mathbf{z}_0)$ using the decoder D.

4 ENVIRONMENT-AWARE LATENT DIFFUSION MODEL (EALDM)

The Environment-Aware Latent Diffusion Model (EALDM) consists of three key stages: encoding, conditioning, and generation. In particular, the encoding process is similar to that in the standard LDM. The encoder E processes the input data $\mathbf{x}^n$ at time n and maps it to a latent space representation $\mathbf{z}^n$. The intuition is for this latent representation $\mathbf{z}^n$ to embed the critical features and dependencies of the input data, which are essential for the generation task. Next, we introduce a UNet-based conditional diffusion model on the obtained latent representation $\mathbf{z}^n$. In particular, The UNet model conditions the generation process for $\mathbf{x}^{n+1}$ on the sequential contextual data until time n, i.e., the temporal context $\mathbf{c}^n$ derived from weather, images, streamflow data, and other time-aware features (e.g., time of day and day of the year). These conditioning factors provide crucial contextual information, which can help effectively guide the generation of future images. Finally, after we obtain the latent representation for the next time step, $\mathbf{z}^{n+1}$, the decoder D then reconstructs the image $\mathbf{x}^{n+1}$ from $\mathbf{z}^{n+1}$. In the following, we will provide details for two major components in the diffusion model: (i) how we incorporate the temporal context as the model condition and (ii) how we leverage the spatial domain information in the diffusion model.

4.1 ENVIRONMENT-AWARE CONDITIONING

Temporal changes in stream images could be determined by weather, water conditions, as well as the time when images are taken. Therefore, it is important to incorporate these influential factors into the image generative process, which essentially captures the transition of image embeddings from each time n to the next time $n+1$. Specifically, we employ the same encoder from the Latent Diffusion Model to embed the previous image $\mathbf{x}^n$, as $\mathbf{z}^n = E(\mathbf{x}^n)$. To capture the temporal information, we apply a long-short term memory (LSTM) model to embed sequential weather data $\{\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^n\}$ and streamflow data $\{y^1, y^2, ..., y^n\}$, as $\tilde{\mathbf{w}}^n = \text{LSTM}(\mathbf{w}^1, \ldots, \mathbf{w}^n)$ and $\tilde{\mathbf{y}}^n = \text{LSTM}(y^1, \ldots, y^n)$.

Additionally, the background and appearance of stream images reflect both seasonal patterns (e.g., green foliage in summer and snow-covered landscapes in winter, as shown in Figure 3) and short-term variations throughout the day (e.g., daytime versus nighttime images). To capture temporal variations in the images, we introduce two variables: t_d, representing the time of day, and t_y, representing the day of the year. The incorporation of these variables into the generative process helps capture both long-term and short-term temporal changes of the environment. Inspired by

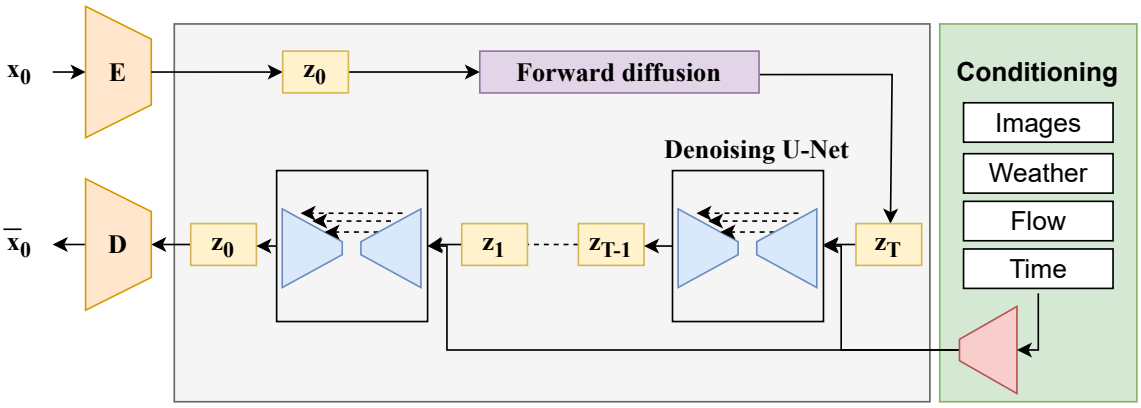


Figure 1: The overall structure of the proposed EALDM method.

(Härkönen et al., 2022), we embed time using sine and cosine functions to preserve its periodic nature, as follows:

$$\tilde{\mathbf{t}}^n(t_d, t_y) = \begin{bmatrix} sin(2\pi\tau_0 t_d) \\ cos(2\pi\tau_0 t_d) \\ sin(2\pi\tau_1 t_y) \\ cos(2\pi\tau_1 t_y) \end{bmatrix} \tag{1}$$

where τ_0 is the day cycle, which is set to 1, and $\tau_1 = \frac{\tau_0}{365.25}$ is the year cycle.

Since weather, streamflow, and time influence temporal changes in stream images, we employ adaptive instance normalization (AdaIN) (Huang & Belongie, 2017) to integrate these factors into the image embedding $\mathbf{z}^n$. Specifically, we used AdaIN to separately integrate each of these factors ($\tilde{\mathbf{w}}^n, \tilde{\mathbf{y}}^n, \tilde{\mathbf{t}}^n$) into the image embedding, and then concatenate the AdaIn outputs with the image embedding, as $\mathbf{h}^n_{\text{AdaIN}} = \left[\text{AdaIN}(\mathbf{z}^n, \tilde{\mathbf{t}}^n), \text{AdaIN}(\mathbf{z}^n, \tilde{\mathbf{w}}^n), \text{AdaIN}(\mathbf{z}^n, \tilde{\mathbf{y}}^n), \mathbf{z}^n\right]$. Then we apply a series of transformations to $\mathbf{h}^n_{\text{AdaIN}}$ with convolutional layers, a bath normalization layer, and a residual layer, as shown in Fig. 2. This process can be expressed as $\mathbf{h}^n_{\text{output}} = \text{Conv} \circ \text{BN} \circ \text{Conv}(\mathbf{h}^n_{\text{AdaIN}}) + \mathbf{z}^n$. Finally, we flatten the obtained embedding $\mathbf{h}^n_{\text{output}}$ and apply a linear layer to generate the condition vector $\mathbf{c}^n$.

Given the condition vector, we then learn a conditional noise prediction at the t^{th} step, as $\boldsymbol{\epsilon}_\theta(\mathbf{z}^n_t, t, \mathbf{c}^n)$, in addition to the original unconditional noise prediction $\boldsymbol{\epsilon}_\theta(\mathbf{z}^n_t, t)$. Following the prior work (Ho & Salimans, 2022), we combine these predictions during inference with a scale factor s, as follows:

$$\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{z}^n_t, t, c) = \boldsymbol{\epsilon}_\theta(\mathbf{z}^n_t, t) + s \cdot (\boldsymbol{\epsilon}_\theta(\mathbf{z}^n_t, t, \mathbf{c}^n) - \boldsymbol{\epsilon}_\theta(\mathbf{z}^n_t, t)) \,. \tag{2}$$

The training loss combines both conditional and unconditional terms: $\mathcal{L} = \mathbb{E}_{t,\mathbf{z}^n_0,\boldsymbol{\epsilon},\mathbf{c}^n}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}^n_t, t, \mathbf{c}^n)\|^2\right] + \lambda\mathbb{E}_{t,\mathbf{z}^n_0,\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}^n_t, t)\|^2\right]$, where λ balances the two components.

4.2 Domain-Discriminative Learning

The environment-aware conditioning ensures that the model generate temporally consistent images that reflect environmental and time changes. A challenge arises as the image-generative model needs to adapt to different spatial domains (i.e., different stream sites) while preserving the domain-specific conditions when used to generate images for a target domain. Our goal is to guide the model to generate accurate images based on the domain-specific conditions while discouraging the influence of other domain conditions. To achieve this, we extend the proposed model by incorporating a domain-discriminative learning method, which helps better differentiate between the conditions of the target domain and those from other domains.

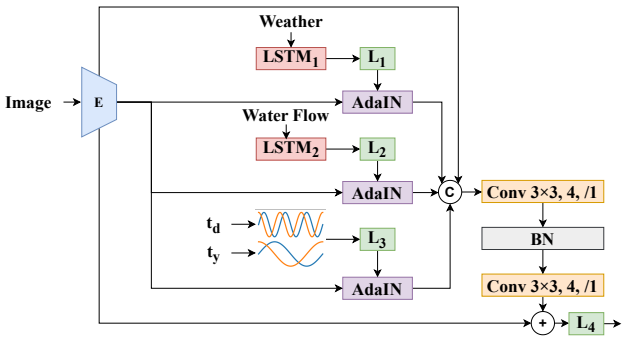


Figure 2: Conditioning architecture.

Specifically, we represent the condition vector for the target domain i at time n as $\mathbf{c}_i^n$, which is obtained using the method described in Section 4.1, and the condition vector for another domain as $\mathbf{c}_j^n$, for $j \neq i$. These condition vectors embed the environmental information needed to capture temporal changes to generate the next stream image at time $n+1$. Using these condition vectors, we generate three versions of predicted noise: $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t^n, t)$ represents the model's predicted noise for $\mathbf{z}_t^n$ at time n and diffusion step t without any domain-specific conditioning, $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t^n, t, \mathbf{c}_i^n)$ denotes the model's predicted noise conditioned on the information of target domain ($\mathbf{c}_i^n$) at time n, and $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t^n, t, \mathbf{c}_j^n)$ is the model's predicted noise conditioned on the information of another randomly sampled domain j ($\mathbf{c}_j^n$) at time n. Combining these predicted noises, we extend Eq. 2 to generate a domain-discriminative noise prediction, $\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_t^n, t, \mathbf{c}_i^n, \mathbf{c}_j^n)$, which is defined as:

$$\hat{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_t^n, t, \mathbf{c}_i^n, \mathbf{c}_j^n) = \boldsymbol{\epsilon}_\theta(\mathbf{z}_t^n, t) + s \cdot (\boldsymbol{\epsilon}_\theta(\mathbf{z}_t^n, t, \mathbf{c}_i^n) - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t^n, t)) + s \cdot \big(\boldsymbol{\epsilon}_\theta(\mathbf{z}_t^n, t, \mathbf{c}_i^n) - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t^n, t, \mathbf{c}_j^n)\big) \tag{3}$$

According to this equation, the domain-discriminative noise leverages both the domain-specific conditions and the differences between the target and other domains. At each time n, the term $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t^n, t)$ provides the baseline prediction for the image $\mathbf{z}_t^n$ at the denoising step t. The first adjustment term $s \cdot (\boldsymbol{\epsilon}_\theta(\mathbf{z}_t^n, t, \mathbf{c}_i^n) - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t^n, t))$ shifts the prediction towards the conditions of the target domain $\mathbf{c}_i^n$. The second adjustment term $s \cdot \big(\boldsymbol{\epsilon}_\theta(\mathbf{z}_t^n, t, \mathbf{c}_i^n) - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t^n, t, \mathbf{c}_j^n)\big)$ introduces a contrast between the target domain $\mathbf{c}_i^n$ and another domain $\mathbf{c}_j^n$, which enforces the model awareness of the background and style differences across domains. By using the domain-discriminative learning, our model benefits from enhanced adaptability and robustness, better aligning with the desired domain characteristics and generating more accurate predictions under varied conditions.

5 Experimental results

5.1 Dataset and metrics

The dataset includes sequential images of nine stream sites located in the West Brook area in western Massachusetts, United States, as well as the local weather and flow data from 2018 to 2020. The dataset, as presented in Table 1, integrates multiple data modalities including high-resolution images, streamflow measurements, and timestamped weather conditions. Table 2, shows the weather data features that are crucial for precise alignment between the generated images and real-world environmental changes. All the stream sites involved in our tests are small streams from the same region. They share the same weather data since they fall within the same grid of existing weather dataset, such as Daymet (day, 2021). The dataset can be accessed through the link `https://github.com/Anonymous/EALDM/data`.

Table 1: Statistics of data from multiple stream sites.

Sites	Train			Validation			Test		
	Size	Flow	Date	Size	Flow	Date	Size	Flow	Date
West Brook Reservoir	271	1.01-76.00	2021/03/25-2021/10/17	108	1.30-122.91	2021/08/21-2021/09/30	162	1.20-147.77	2021/07/06-2021/08/21
West Brook Lower	680	0.20-373.18	2019/12/31-2021/11/01	272	2.86-363.90	2021/07/25-2021/09/24	480	1.42-373.00	2021/04/25-2021/07/25
West Brook Upper	788	0.04-263.43	2020/01/03-2021/11/06	315	0.58-196.98	2021/06/13-2021/08/22	472	1.11-122.81	2021/02/13-2021/06/13
Avery Brook Right	540	0.77-69.59	2021/03/19-2021/12/21	216	2.24-76.94	2021/10/03-2021/11/21	323	1.86-283.25	2021/07/14-2021/10/03
Avery Brook Left	366	1.78-283.25	2021/07/02-2021/12/21	146	4.45-73.00	2021/10/29-2021/12/01	220	2.04-76.94	2021/09/08-2021/10/29
Avery Brook Bridge	155	2.44-57.62	2021/03/10-2021/12/21	38	3.83-66.77	2021/08/04-2021/11/27	39	0.81-61.49	2021/06/07-2021/08/04
Obear Brook Lower	320	0.03-13.16	2021/03/30-2021/11/01	128	0.03-13.27	2021/07/09-2021/10/14	193	0.01-4.69	2021/05/23-2021/07/09
West Whately	137	0.47-6.66	2021/04/06-2021/10/28	35	0.65-10.55	2021/07/31-2021/10/06	35	0.12-4.90	2021/06/09-2021/07/31
Sanderson Brook	168	1.19-38.37	2021/04/01-2021/10/28	45	1.49-38.52	2021/08/21-2021/10/02	42	1.03-28.70	2021/06/19-2021/08/21

Table 2: Summary statistics for weather data.

Statistic	Average Temp (C°)	Max Daily Temp (C°)	Min Temp (C°)	Wind Speed (m/s)	Wind Direction (°)	Max Wind Speed (m/s)	Mean Relative Humidity (%)	Atmospheric Pressure (mb)	Mean Solar Radiation (W/m^2)	Total Rainfall (mm)
Mean	9.40	14.81	4.40	1.45	217.91	9.17	72.61	1017.39	156.59	3.74
Std	9.98	10.73	9.91	0.67	104.81	3.71	18.88	7.28	94.61	9.19
Min	-18.92	-17.18	-49.49	0.00	0.00	0.00	20.77	988.00	5.02	0.00
Max	29.08	36.03	24.28	5.37	359.90	27.93	100.00	1039.00	365.70	84.60

We aim to determine whether the generated images resemble the original images in terms of their visual quality. To achieve this, we use the Fréchet Inception Distance (FID) score (Heusel et al., 2017) to measure the distance between the distributions of generated images and real images in their feature space.

Second, we evaluate whether the conditional information has been effectively incorporated into the generated images. Given the complexity of the conditional information, direct labeling is not straightforward. Instead, we first utilize GPT-4 (OpenAI, 2023) to classify the weather data of the subsequent day $n+1$ into following commonly occurring

weather labels: "Sunny/Clear", "Cloudy/Overcast", "Rainy", "Snowy", "Foggy/Misty", "Windy", "Stormy/Severe", "Hot/Heatwave", "Cold/Cold Wave", "Mixed/Variable". We then use ResNet to classify the generated images into these weather categories. We perform binary classification for each weather category and measure the average F-1 score over different categories.

Furthermore, CLIP (Radford et al., 2021) is employed to assess the alignment between the generated images and the ground truth images. We use CLIP to classify both the generated and ground truth images into the same nine weather categories. By comparing the classified labels of the generated images with those of the ground truth, we evaluate whether the generated images exhibit the same environmental conditions as shown in the true images. This helps verify that the conditional information has been effectively integrated into the generation process. The accuracy is computed as the fraction of weather categories that share the same outputs between generated and true images, as $\text{Accuracy} = \frac{\sum(\text{CLIP}(\hat{x})==\text{CLIP}(x))}{|\#\text{Categories}|}$, where $\text{CLIP}(\hat{x})$ is the classification output of the generated image for different weather categories, and $\text{CLIP}(x)$ is the classification output of the ground truth image. We report the average CLIP accuracy over all the test samples.

To evaluate the realism of the generated images, we use ResNet to predict water flow (in cubic feet per second) at each time. We measure the mean squared error between the predicted and observed water flow values. A lower error indicates that the generated images more accurately reflect the realistic water flow patterns, thus demonstrating a higher degree of realism of the generated images.

5.2 Experiment Setup

The proposed method is implemented using PyTorch, and trained on a single NVIDIA RTX A6000 GPU with 48 GB memory. We use VQGAN-f-8 for first stage auto encoder part. Image resolution for first stage part is 256^2, z-shape is $4 * 32 * 32$, model-channels are 256, the transformer depth is 1, and the number of heads is 32. The conditioning is implemented by cross attention to integrate c and z in latent diffusion, with condition dimension of 512. The diffusion channel multiplier is set to $\{1, 2, 4\}$, the batch size is 4, the length of sequence is 8, and the learning rate is $1e-06$. The scale factor s is set to 2.

5.3 Quantitative Analysis of Model Performance

We chose Latent Diffusion Models (LDMs), Classifier-Free Diffusion Guidance (CFDG), BLIP-Diffusion (Li et al., 2024), and ControlNet (Zhang et al., 2023) as baselines due to their complementary strengths in image generation tasks conditioned on time and environmental factors. LDMs operate efficiently in the latent space of pre-trained autoencoders, which allows for high-quality image generation while maintaining computational efficiency. This is crucial for handling complex conditioning inputs like weather and time. Additionally, CFDG improves control over the generation process by leveraging both the latent representation and conditioning input, providing sharper and more accurate results that reflect the specific climate conditions being modeled. Moreover, BLIP-Diffusion enhances the model's ability to interpret and generate images based on detailed textual descriptions, such as complex environmental and climate factors, by aligning vision and language representations effectively. ControlNet offers fine-grained control over the generated images by conditioning on structural inputs like depth maps, ensuring that the spatial coherence of the scene is preserved while adjusting for changes in environmental conditions over time. Comparison with these methods can help validate that our model not only generates high-quality images but also adapts accurately to environment-related variations.

In Table 3, we present the performance of different methods. In our experiments, we investigated two sampling methods for handling the time-series nature of the data: random splitting and sequential splitting . The sequential splitting uses the first (60%) time period for training, (20%) time period for validating the remaining (20%) time period for testing. In contrast, the random splitting randomly picks (60%) samples for training. As expected, our results indicate that random splitting yields better performance compared to sequential splitting. This improvement can be attributed to the fact that random splitting provides a more diverse and representative training samples, and their distribution is closer to that of the test data. An interesting observation from Table 3 is that the GPT score is consistently lower than the CLIP accuracy. The dataset contains many night-time images, and it is challenging to determine weather information from images at night. The proposed model, EALDM, shows the strongest overall performance, particularly when incorporating all features (image, weather, flow, time), with the highest CLIP score and GPT F-score, less flow prediction error and competitive FID errors under both sequential and random splits. BLIP-Diffusion and ControlNet struggle with CLIP score FID, indicating limitations in image quality. LDM excels in FID but performs poorly on other measures, highlighting its focus on visual realism at the expense of content alignment.

Table 3 also presents a comparison of how different conditioning factors—image, weather, flow, and time—affect the performance of the model across various evaluation metrics. We can observe that the CLIP alignment between the generated images and the ground truth (subsequent images) improves with each added condition. This suggests that incorporating more conditioning information helps better capture the intended weather and time-related context. we also report the accuracy of weather classification from generated images, using a ResNet model to predict the weather conditions based on the image outputs. The ground truth weather labels are derived using GPT-4 to classify the conditioning data into one of nine common weather categories. This metric evaluates how well the generated images reflect the intended weather conditions, as conveyed through the real weather data on the subsequent day. We observe that as more conditions are added (image, weather, flow, time), the accuracy of these weather classifications improves, indicating that the model is successfully incorporating complex conditional information into the generated outputs. The model's ability to predict water flow also gets improved as more conditions are added. This is reflected in a lower mean squared error (MSE) for water flow predictions, showing that the generated images align more closely with real-world physical dynamics, such as water movement, when conditioned on additional information. However, it can be seen that adding each condition tends to slightly increase the Fréchet Inception Distance (FID). This is expected, as conditioning on more factors adds complexity to the generation process, making it harder to precisely match real images. This study highlights the trade-off between visual quality (FID) and the alignment of the generated images with the semantics of conditioning information (CLIP accuracy, GPT label accuracy, and water flow realism).

The results also reveal the effect of domain-discriminative learning on model performance, by comparing EALDM and EALDM no DG in Table 3. While incorporating domain-discriminative learning leads to a slightly increase in FID, it brings improvements in other metrics. These improvements occur because the domain guidance helps the model focus on and leverage domain-specific features and constraints, thereby enhancing its ability to generate more contextually relevant and accurate outputs tailored to the specific conditions.

Table 3: Comparison of model performances in image generation. The table highlights the evaluation of several models based on CLIP alignment, GPT-labeled weather classification, FID, and flow prediction errors. It also shows the effects of sequential and random data splitting and domain guidance. "EALDM no DG" represents the variant of the proposed method without using the domain-discriminative learning.

Model	Image	Weather	Flow	Time	Sequential				Random			
					CLIP	GPT F-score	FID	Flow Error	CLIP	GPT F-score	FID	Flow Error
BLIP-Diffusion	✓	✓			0.05	0.43	6.20	6.2	0.11	0.49	5.72	4.5
ControlNet	✓	✓			0.18	0.30	4.92	4.9	0.24	0.40	13.51	4.0
LDM					0.08	0.05	**0.41**	5.1	0.10	0.06	**0.30**	4.2
CFDG	✓				0.55	0.42	0.53	3.17	0.63	0.58	0.36	1.76
EALDM	✓			✓	0.58	0.42	0.47	2.64	0.68	0.65	0.37	1.48
EALDM	✓	✓		✓	0.51	0.43	0.48	2.67	0.79	0.72	0.38	1.46
EALDM	✓		✓	✓	0.55	0.40	0.48	2.57	0.71	0.67	0.39	1.41
EALDM	✓	✓	✓	✓	**0.63**	**0.53**	0.89	**2.16**	**0.84**	**0.75**	0.48	**1.37**
EALDM no DG	✓	✓	✓	✓	0.61	0.50	0.45	2.54	0.82	0.71	0.32	1.41

5.4 Qualitative Evaluation of Generated Samples

Figure 3 shows examples of images generated by our model in comparison with BLIP-Dissusion and ControlNet. Although BLIP-Diffusion integrates weather conditions, it fails to accurately maintain the structure and content related to the current timestamp, leading to inconsistencies between the generated images and the intended scene details. ControlNet, while better at adhering to image structure and content, struggles with incorporating weather effects accurately and often results in lower-quality images. In contrast, our model integrates all specified conditions—current image, weather, timestamp, and water flow—ensuring that the generated images are both structurally and semantically accurate.

Figure 4 illustrates the results of an ablation study comparing image generation with and without the incorporation of weather and water flow conditions. The samples highlight how including these conditions affects the generated images alignment with the subsequent image in the ground truth. In the first row, the ground truth shows an increased water level that obscures rocks in the river. This detail is accurately reflected in the generated image when weather and water flow conditions are included but is missed when these conditions are not incorporated. Similarly, in the second row, the ground truth depicts a decrease in water flow revealing more rocks, a change that is accurately captured only when the conditions are considered. In the third and fourth rows, incorporating these conditions allows the model to reflect changes in the scene perfectly, including generating a snowy scene when weather conditions are taken into account. Without these conditions, the model fails to generate the snowy environment, demonstrating how crucial weather and water flow conditions are for generating accurate and contextually relevant images. These examples underscore the

importance of incorporating weather and water flow conditions to achieve realistic and contextually accurate image generation, as neglecting these factors results in less accurate representations of scene changes. We also show the flow predictions by the proposed method vs. CFDG in Figure 5, which confirms that the images generatd by our method can effectively preserve the semantics of water dynamics.

In Figure 6, we present examples comparing the performance of our model with and without domain-discriminative learning. Our findings highlight that the model without domain-discriminative learning occasionally fails to accurately reflect the structure and content of images from specific sites. This issue arises because the training data does not cover the entire year for some sites, preventing the model from fully capturing seasonal patterns. As a result, the model tends to relate the timestamps to other sites it has already seen, leading to generated images that resemble previously observed structures and content for the timestamp. The proposed domain-discriminative learning addresses this problem by providing additional context and constraints specific to the target domain, thereby improving the model's ability to generate images that are consistent with the target site's characteristics. As shown in the examples, when the domain-discriminative learning is applied, the generated images align more closely with the target site, demonstrating enhanced accuracy and relevance in reflecting the target site's structure and content.

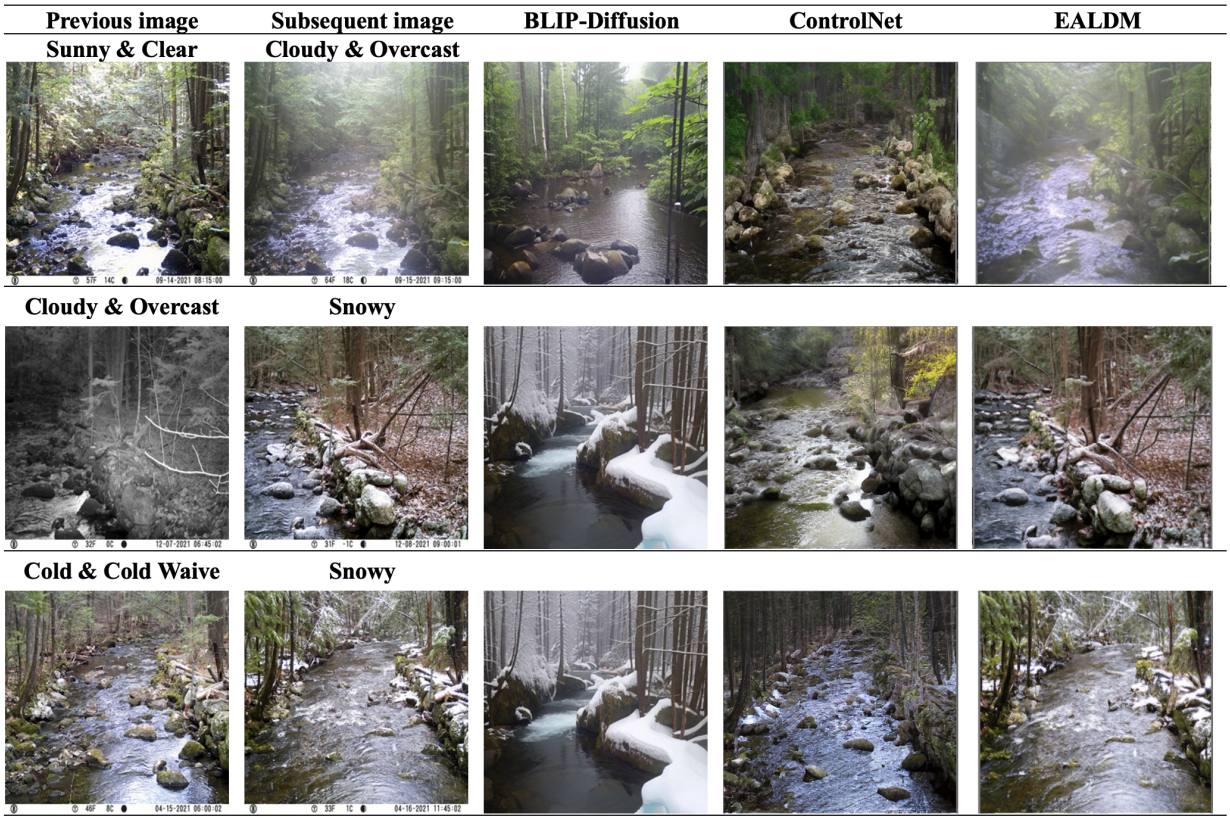


Figure 3: Comparison of images generated by different methods (columns 3-5) and the ground truth images (2^{nd} column).

6 Conclusion

In this work, we introduced the Environment-Aware Latent Diffusion Model (EALDM), which integrates multimodal data such as weather, streamflow values, and temporal information into the generative process to produce realistic, environment-consistent images over time. EALDM significantly expands the capabilities of latent diffusion models by conditioning the generation process on temporal and environmental context. New model architectures have been developed to effectively embed contextual information from different sources. Our method not only predicts future visual changes based on past conditions but also ensures that the generated images precisely reflect realistic weather, time of day, and seasonal variations for the target stream site.

We demonstrated the effectiveness of EALDM through extensive evaluations, showing that it can generate temporally coherent images that align with the environmental context. The inclusion of domain-discriminative learning further improves the adaptability of our model to specific sites, enabling more accurate image generation at different locations. The results highlight the potential of EALDM for applications in climate-aware image generation, environmental monitoring, and temporal forecasting.

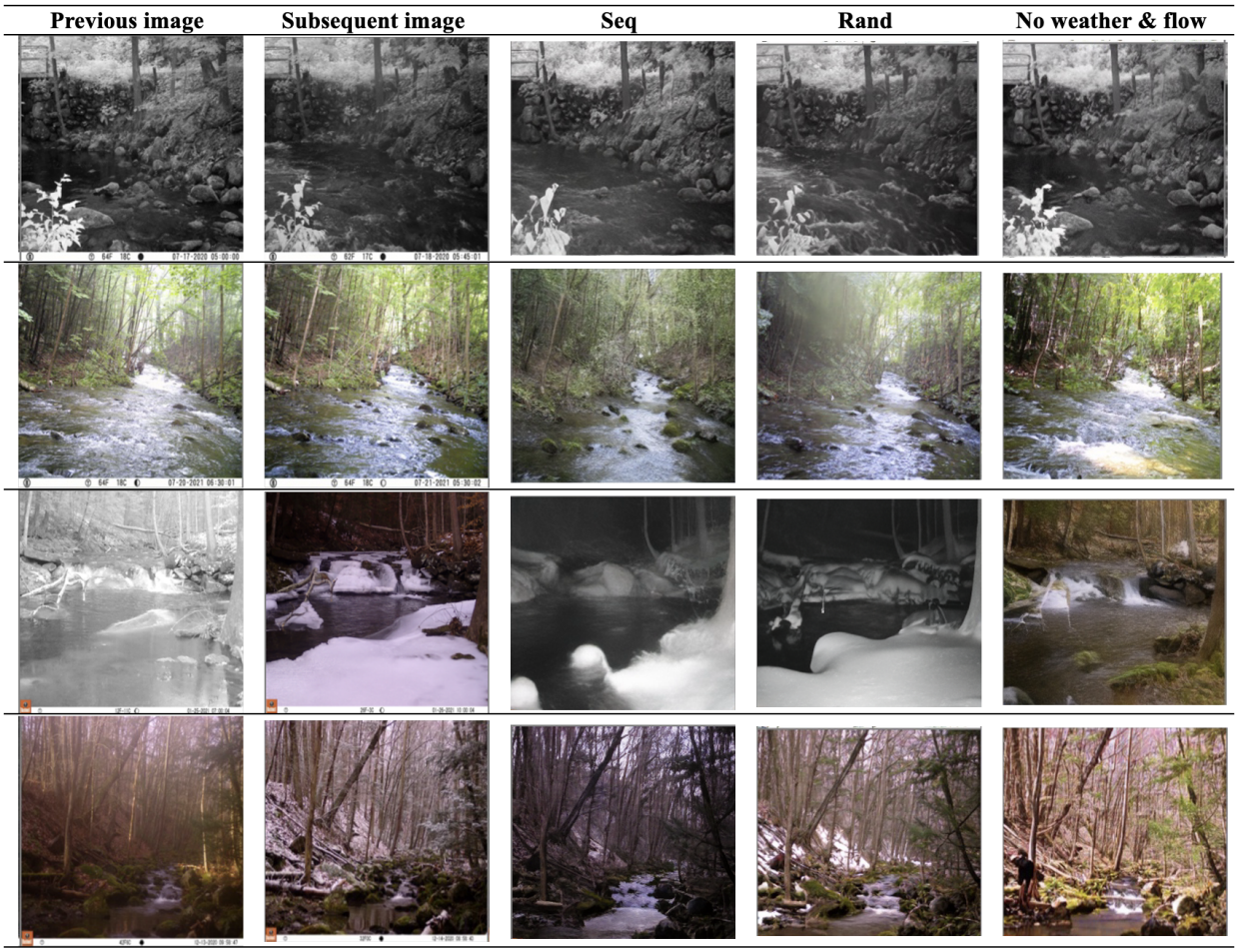
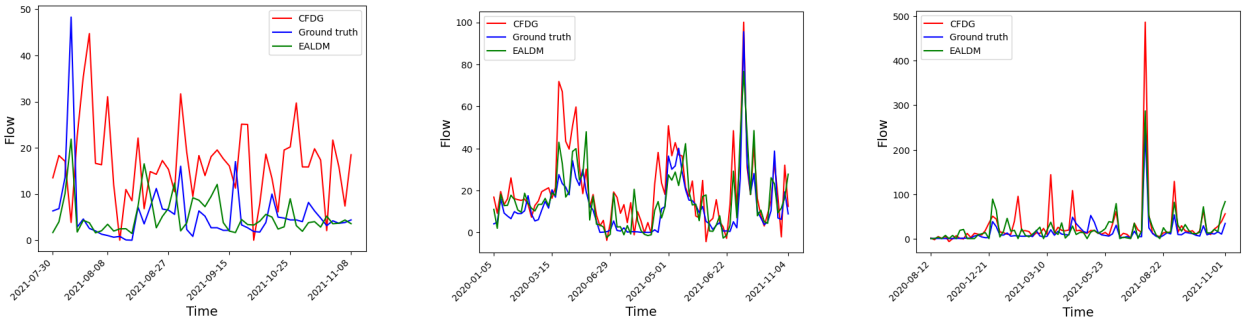


Figure 4: Comparison of sequential data splitting, random data splitting, and sequential splitting without using the conditions of weather and water flow.



(a) Avery Brook Right (b) West Brook Upper (c) West Brook Lower

Figure 5: Comparison of flow predictions between EALDM and CFDG across different stream sites.

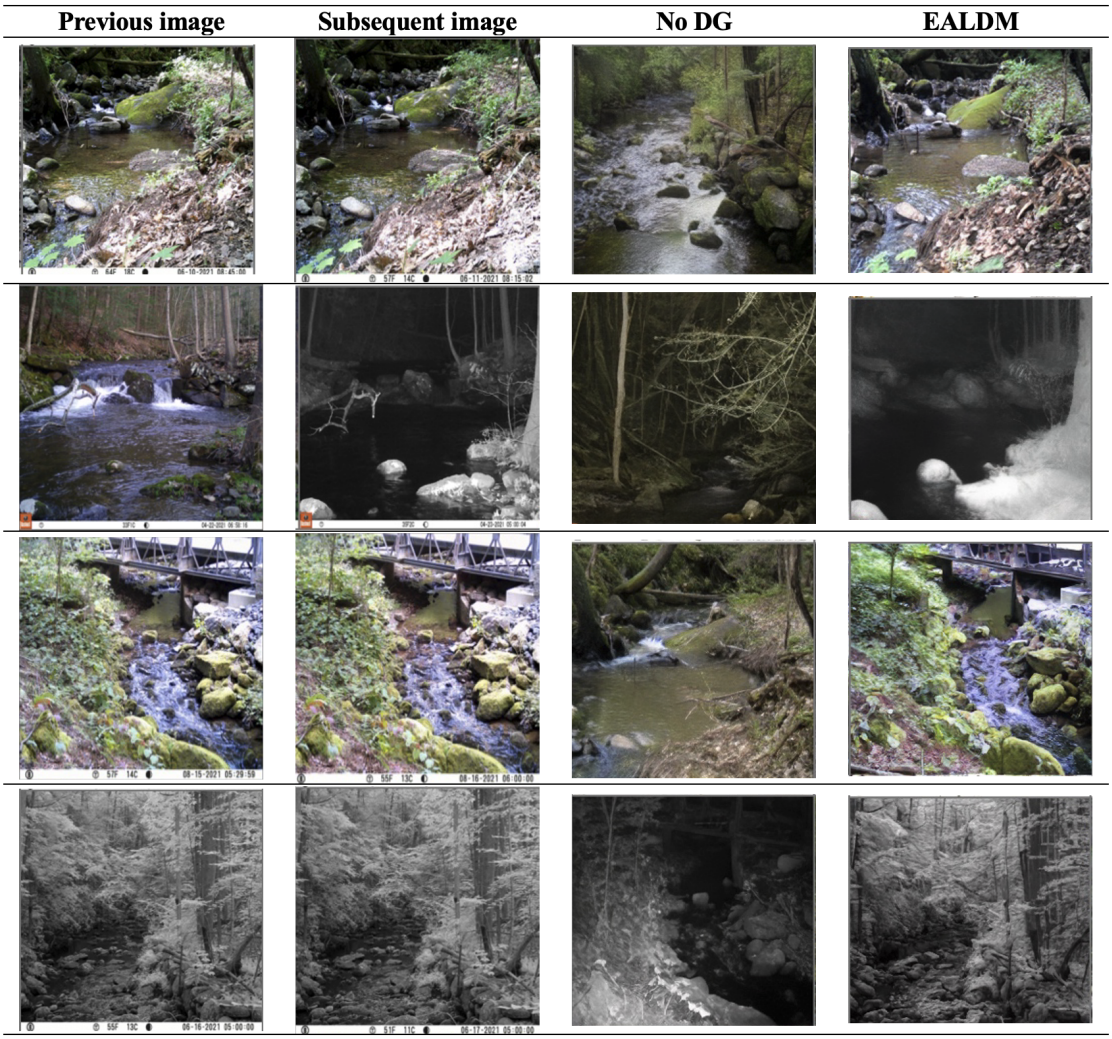


Figure 6: Comparison of using domain guidance information (i.e., domain-discriminative learning).

7 Limitations and Future works

While the proposed EALDM provides significant advancements in temporally conditioned image generation, there are several areas for future exploration. First, incorporating higher spatial-temporal resolutions could enhance the accuracy of environment-dependent predictions, especially in regions where small-scale variations, such as microclimates, play a significant role. Additionally, integrating more diverse data sources, such as satellite imagery or real-time IoT sensor data, would further enrich the conditioning process, enabling more precise and real-time predictive capabilities. Another promising avenue is the extension of this work to 3D or video-based temporal generation, which could better capture dynamic environmental changes. Finally, exploring more sophisticated learning frameworks, such as reinforcement learning or adversarial training techniques, could further refine the model's predictive capabilities.

Despite its effectiveness, EALDM has some limitations. First, the model's reliance on multi-modal data sources introduces challenges related to data availability, quality, and synchronization. Inconsistent or missing environmental data could degrade model performance, particularly in real-world applications. Additionally, the current framework assumes static relationships between environmental variables, which may not fully capture complex interdependencies like feedback loops or long-term climatic shifts. Another limitation is the computational overhead, as the training process for high-resolution, environment-aware diffusion models can be resource-intensive, making scaling to larger datasets or more complex scenarios costly. Finally, the model's capacity to generalize to unseen environments remains a challenge, particularly in regions where the conditions are drastically different from the training data.

REFERENCES

U.s. geological survey, webcams. URL https://www.usgs.gov/products/multimediagallery/webcams.

Daymet. https://daac.ornl.gov/cgi-bin/dataset_lister.pl?p=32, 2021.

Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18370–18380, 2023.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18392–18402, 2023.

Pheno Cam. The phenocam network. URL https://phenocam.sr.unh.edu/webcam.

Hikmet Kerem Cigizoglu. Application of generalized regression neural networks to intermittent flow forecasting and estimation. *Journal of Hydrologic Engineering*, 2005.

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

Dapeng Feng et al. Enhancing streamflow forecast and extracting insights using long-short term memory networks with data integration at continental scales. *WRR*, 2020.

Rahul Ghosh, Arvind Renganathan, Kshitij Tayal, Xiang Li, Ankush Khandelwal, Xiaowei Jia, Christopher Duffy, John Nieber, and Vipin Kumar. Robust inverse framework using knowledge-guided self-supervised learning: An application to hydrology. In *SIGKDD*, 2022.

Erik Härkönen, Miika Aittala, Tuomas Kynkäänniemi, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Disentangling random and cyclic effects in time-lapse sequences. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 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.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1501–1510, 2017.

Xiaowei Jia, Jacob Zwart, Jeffrey Sadler, Alison Appling, Samantha Oliver, Steven Markstrom, Jared Willard, Shaoming Xu, Michael Steinbach, Jordan Read, et al. Physics-guided recurrent graph model for predicting flow and temperature in river networks. In *SDM*. SIAM, 2021.

Xuhui Jia, Yang Zhao, Kelvin CK Chan, Yandong Li, Han Zhang, Boqing Gong, Tingbo Hou, Huisheng Wang, and Yu-Chuan Su. Taming encoder for zero fine-tuning image customization with text-to-image diffusion models. *arXiv preprint arXiv:2304.02642*, 2023.

Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv:1710.11431*, 2017.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6007–6017, 2023.

Dongxu Li, Junnan Li, and Steven Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. *Advances in Neural Information Processing Systems*, 36, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023a.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22511–22521, 2023b.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

OpenAI. Gpt-4 technical report, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22500–22510, 2023.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

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

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023.

Xiaohu Zhao, Kebin Jia, Benjamin Letcher, Jennifer Fair, and Xiaowei Jia. Bringing vision to climate: A hierarchical model for water depth monitoring in headwater streams. *Information Fusion*, 110:102448, 2024.