

# Robot Shape and Location Retention in Video Generation Using Diffusion Models

Peng Wang, Zhihao Guo, Abdul Latheef Sait, Minh Huy Pham

**Abstract**—Diffusion models have marked a significant milestone in the enhancement of image and video generation technologies. However, generating videos that precisely retain the shape and location of moving objects such as robots remains a challenge. This paper presents diffusion models specifically tailored to generate videos that accurately maintain the shape and location of mobile robots. The proposed models incorporate techniques such as embedding accessible robot pose information and applying semantic mask regulation within the scalable and efficient ConvNext backbone network. These techniques are designed to refine intermediate outputs, therefore improving the retention performance of shape and location. Through extensive experimentation, our models have demonstrated notable improvements in maintaining the shape and location of different robots, as well as enhancing overall video generation quality, compared to the benchmark diffusion model. Codes will be open-sourced at: <https://github.com/PengPaulWang/diffusion-robots>.

## I. INTRODUCTION

Diffusion models have achieved remarkable advancements in recent years, and have achieved better or on-the-par performance with generative adversarial networks in image and video generation [1], [2]. Compared to image generation, video generation remains a challenge in terms of model complexity, dependence on data and computational resources, consistency of generated videos, generation efficiency, and shape and location retention of dynamic objects in generated videos [3], [4]. Despite all the challenges, the potential of diffusion models to generate dynamic and appealing content has driven the research and application forward, and they have been applied in generating high-quality videos [4], carrying out video prediction and infilling [3], control movements in the generated video [5], directly process and manipulate a real input video [6], and human feature refinement [7]. Another promising application of diffusion models is that they can be used to generate data for robotic applications like human-robot collaboration, where collecting real data for model training faces legal and ethical challenges.

The foundational technology behind many of the applications mentioned is the Denoising Diffusion Probabilistic Model (DDPM), which is trained to understand Gaussian noise patterns added to input images throughout the training process. Once sufficiently trained, the DDPM can start with noisy images or images that consist purely of Gaussian noise

\*This work was supported by the Department of Computing and Mathematics, Manchester Metropolitan University.

Peng Wang (p.wang@mmu.ac.uk), Zhihao Guo, Minh Hui Pham are with the Department of Computing and Mathematics, Manchester Metropolitan University, M15 6BH Manchester, UK. Abdul Latheef Sait is with JD Sports Fashion PLC, and the work was done while at MMU.



Fig. 1: The original and generated frames of a robot. *Left*: the original frame; *Middle*: the frame generated by a proposed model; *Right*: the frame generated by the benchmark model. The robot arm is broken in the frame generated by the benchmark model.

and, through iterative denoising, produce outputs that adhere to a specific empirical distribution [4], [8]–[10].

The evaluation of diffusion models’ performance often relies on metrics like the Peak Signal-to-Noise Ratio (PSNR), which measures the overall quality of frames or videos by computing pixel-to-pixel differences between the generated frames and the reference frames if any. However, relying solely on PSNR may overlook structural information loss, such as local distortions of the shape of objects of interest, providing a misleadingly positive assessment of overall performance. For instance, Figure 1 shows one original frame (*left*) with a robot, and two frames generated by diffusion models (*middle* and *right*). We can see that the generated frame on the right has a broken arm (lost retention of the shape), while the generated frame in the middle maintains the shape of the arm. Despite the failed arm shape retention, the two generated frames have similar PSNR values as the distorted arm does not contribute enough to make a distinctive difference in PSNR values. This oversight is particularly critical in scenarios where an object’s shape and location are crucial in generated frames. For instance, in human-robot collaborative tasks, there is the need to forecast potential collisions between humans and (dynamic) robots, and the collection of such data for collision model training in real life often faces ethical and legal challenges. Therefore, using diffusion models to generate data with shape and location retention becomes a promising solution. In light of these observations, the Structural Similarity Index (SSIM) emerges as an alternative metric for evaluating diffusion models. Unlike PSNR, SSIM is adept at capturing structural similarities and differences, making it a more reliable indicator of a model’s ability to preserve object shapes and locations.

This paper aims at developing diffusion models that can generate frames where the shape and location of objects of interest can be retained. Particularly, we are interested in generating videos that contain moving robots, whose shape

and location retention are vital in the generated frames. As mentioned earlier, this holds the potential to generate data for human-robot collaboration model training and bypass legal and ethical hurdles. Two types of robots are used in different scenarios, i.e., a Waffle Pi mobile robot with a gripper mounted on top and a collaborative robot, a.k.a., cobot. The proposed diffusion models take the ConvNext [11] as the backbone network, to accelerate the training and sampling efficiency [6]. To retain the shape and location of the robots, we have embedded the robot pose information such as location, orientation, and velocities into ConvNext blocks and used semantic masks (either the masks of the robots or the masks of the robots and the backgrounds) to regulate the intermediate outputs of ConvNext blocks. Various experiments have been conducted to investigate how pose embedding and mask regulation affect the performance of the models in shape and location retention.

The contributions of this work include 1) the development of diffusion models capable of preserving the shape and location of robots within generated frames. 2) the introduction of a novel Spatially-Adaptive Normalization (SPADE) module for integrating semantic masks, and the implementation of an embedding procedure that incorporates robot pose information from controllers like the Robot Operating System (ROS) into the backbone network, which strikes a balance between the quality of generation and the preservation of shape and location information. 3) Introduction of a refined Intersection over Union (IoU) metric and the Hu moments match for evaluating the retention of location and shape.

The remainder of the paper is organised as follows: Section II presents some related works, Section III elaborates on the approach, Section IV covers experiments, discussions, and an ablation study, and finally, Section V concludes the paper.

## II. RELATED WORKS

Most video generation models based on DDPMs share a common underlying core backbone, specifically the UNet architecture [4], [12], which is utilized for the denoising process. However, these models differ significantly in the conditions they employ for generating new frames. Broadly, these conditions can be categorized into three types:

- **Embedded Context Information:** For example, Yang et al. [13] introduce residual video diffusion, where a context vector, generated by a convolutional recurrent neural network, is used as a condition to generate the next frame.
- **Semantic Masks:** Wang et al. [10] propose the Semantic Diffusion Model, which employs semantic masks to condition the generation of new frames, particularly improving the quality of small objects in the video.
- **Video Frames as Conditions:** Vikram et al. [14] propose masked conditional video diffusion, where certain frames from the past or future are masked. The model is then trained on unmasked frames and generates the masked frames based on a predefined masking strategy.

Yaniv et al. [6] recently introduced SinFusion, a video generation diffusion model that leverages ConvNext [11] as its backbone. ConvNext, a pure ConvNet architecture, has demonstrated equivalent or superior performance compared to Transformers in terms of accuracy and scalability, particularly in detection and segmentation tasks on datasets like ImageNet and COCO. This makes ConvNext a strong candidate for enhancing the efficiency of diffusion models. SinFusion exploits the strengths, offering significant advantages in training DDPMs on a single image or its large crops, effectively addressing the overfitting issues typically associated with using UNet as a backbone.

The authors have identified potential drawbacks of SinFusion, such as the generation of distorted dynamic objects, as illustrated in Figure 1 (*Right*). This issue highlights the importance of shape retention in diffusion-based generative models. Additionally, it has been observed that generative models may struggle to maintain the correct location of dynamic objects in the generated outputs. In applications like human-robot collaboration [15], the retention of both shape and location is crucial for generating accurate data for downstream model training.

Several approaches have been proposed to address the challenge of shape retention in diffusion models. For example, Okuyama et al. [7] utilize diffusion models to refine facial features after human pose and body editing. Similarly, Holmquist et al. [16] apply diffusion models to recover 3D human poses from single images. These studies focus on human generation and refinement, with less emphasis on location retention. Our work, however, focuses on a robotic context, prioritizing the accurate generation of dynamic robots while maintaining both shape and location integrity.

## III. DIFFUSION MODELS

The theory and fundamental principles of diffusion models were introduced by Sohl-Dickstein et al. [8] and further elaborated upon in subsequent studies like those by Ho et al. [4], [9], as well as other works such as that by Hoppe et al. [3]. In essence, diffusion models utilize a deep neural network  $\mathcal{M}$ , such as UNet [12], as their backbone network. This network is trained on noisy data, such as images and video frames, to enable the trained model to accurately identify and model the noise present in the input data.

The training of diffusion models comprises two primary stages: the forward diffusion process (forward process) and the reverse diffusion process (reverse process). In the forward process, data such as images and videos serve as inputs, and the structure of the data distribution is disrupted by introducing noise. This facilitates the training of model  $\mathcal{M}$  to recognize and model the noise imposed on the data. The reverse diffusion process, known as the reverse process, aims to reconstruct the data structure from noisy data or the noise itself. In this paper, we will first review these two stages of diffusion models in the context of video generation, followed by our proposed works.

### A. The Forward Diffusion Process

In the context of image/video generation, given an input frame  $\mathbf{x}_0$  sampled from a distribution  $q(\mathbf{x}_0)$ , one can iteratively add Gaussian noise  $\Sigma_t \sim \mathcal{N}(\Sigma_t; \mathbf{0}, \mathbf{I})$ ,  $t = 1, \dots, T$  to  $\mathbf{x}_0$  for  $T$  steps. This process generates a sequence of noisy samples  $\{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ . The variance of the noise added at each step can be controlled using a variance scheduler  $\{\beta_t \in (0, 1)\}_{t=1}^T$ . The forward diffusion process is normally formulated as a Markov chain:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad (1)$$

where

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \quad (2)$$

which indicates the dependency of  $\mathbf{x}_t$  on  $\mathbf{x}_{t-1}$ . This also implies that to get a noisy sample at  $\mathbf{x}_t$ , one needs to add noises from  $\mathbf{x}_0$  up to  $\mathbf{x}_{t-1}$  step by step, which could be time and computational resources demanding. Fortunately, this can be simplified as shown in [9], i.e., the forward process admits sampling  $\mathbf{x}_t$  at an arbitrary timestep  $t$  in closed form. This is achieved by letting  $\alpha_t = 1 - \beta_t$  and  $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$ , one then gets

$$q(\mathbf{x}_t|\mathbf{x}_0) := \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I}), \quad (3)$$

which indicates that  $\mathbf{x}_t$  can be sampled from  $\mathbf{x}_0$  in one step as in

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\Sigma, \quad (4)$$

where  $\Sigma \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  is the noise used to generate the noisy frame  $\mathbf{x}_t$ .

### B. The Reverse Diffusion Process

The reverse diffusion process involves starting with a Gaussian noise  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  and then reversing the transition outlined in Equation (1). This reversal allows for sampling from the posterior of the forward process  $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ , with  $t = T, \dots, 1$ , in order to recover  $\mathbf{x}_0$  (it's worth noting that the process can terminate at any intermediate stage). However, reversing Equation (1) presents a challenge, and it is typically approximated using a trainable Markov chain depicted in Equation (5), which begins with a Gaussian noise  $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$ :

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad (5)$$

where

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)). \quad (6)$$

One can see that if  $p_\theta(\mathbf{x}_{0:T})$  can be learned by  $\mathcal{M}$ , then the reverse process simplifies to

$$p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}, \quad (7)$$

where  $\mathbf{x}_{1:T}$  are latent variables of the same dimensions with  $\mathbf{x}_0$ . The approximation of  $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$  using  $p_\theta(\mathbf{x}_{0:T})$

is achieved by optimising the variational bound on negative log-likelihood between them [9]:

$$\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q \left[ -\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] := L, \quad (8)$$

which can be rewritten into Equation (9) according to [8]:

$$L := \mathbb{E}_q \left[ \underbrace{D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \| p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^T \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} - \underbrace{\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \right], \quad (9)$$

where  $D_{\text{KL}}$  represents the KL divergence. One can see that each term in Equation (9) is a direct measure of the similarity in terms of KL divergence between  $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$  and the reversed forward transitions but conditioned on  $\mathbf{x}_0$ , i.e.,  $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ . It is noteworthy that  $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$  is tractable and this makes optimisation of  $L$  viable, henceforth making the approximation of  $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$  using  $p_\theta(\mathbf{x}_{0:T})$  viable.

In the context of video generation, an arbitrary noisy sample  $\mathbf{x}_t$ ,  $t = T, \dots, 1$  sampled using Equation (4) is fed to the deep neural network-based model  $\mathcal{M}$ , which is trained (by optimising Equation (9)) to approximate the noise  $\Sigma_t$  imposed. When well trained,  $\mathcal{M}$  will be able to identify and model the noises, helping to remove the noise and restore data structures.

Inspired by advancements in image and video generation, researchers have introduced various diffusion models. These models include those that utilise semantic masks as conditions to produce high-quality images [10], among others. Semantic masks offer valuable information, such as object shapes and locations, making them ideal for generative tasks that prioritise retaining shape and spatial details. Denoting conditions like masks as  $\mathbf{y}$ , Equation (5) can be reformulated as:

$$p_\theta(\mathbf{x}_{0:T}|\mathbf{y}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}), \quad (10)$$

where

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, \mathbf{y}, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, \mathbf{y}, t)). \quad (11)$$

Since the condition  $\mathbf{y}$  applies to  $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y})$  for  $t = T, \dots, 1$ , it is straightforward to substitute these terms involve  $\mathbf{y}$  into Equation (9) to derive the optimization term for

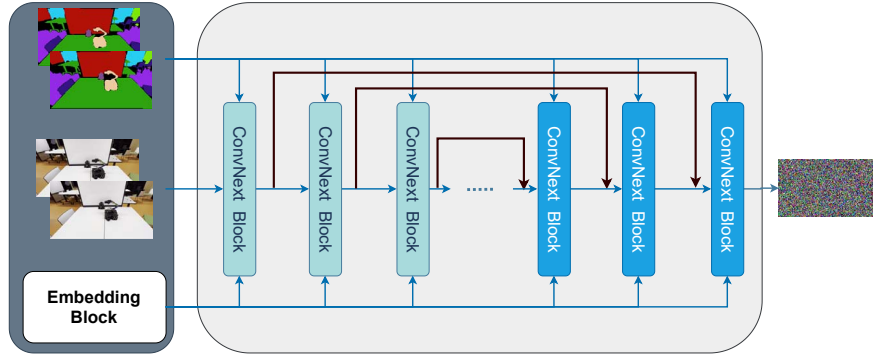


Fig. 2: The overall architecture of the proposed diffusion model for shape and location retention. The black arrows indicate residual connections. It is worth noting we use images that depict masks of both the robot and the background. However, we also consider cases where only robot masks are used in this paper.

conditioned diffusion models:

$$L = \mathbb{E}_q \left[ \underbrace{D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_0) \| p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^T \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{y}))}_{L_{t-1}} - \underbrace{\log p_\theta(\mathbf{x}_0 | \mathbf{x}_1, \mathbf{y})}_{L_0} \right]. \quad (12)$$

When the model is well trained, it will take in a Gaussian noise image  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  and ‘recreate’ samples from it by removing the noise step by step.

#### IV. SHAPE AND LOCATION RETENTION DIFFUSION MODELS

##### A. Overall Architecture

Figure 2 shows the overall architecture of the proposed shape and location retaining diffusion models, as well as the inputs and outputs of the model. We have adopted the ConvNext [11] as the backbone network. We have introduced semantic mask regulation and robot pose embedding into the module, to improve shape and location retention performance. The mask regulation and robot pose embedding modules are depicted in Figure 3 and Figure 4, respectively. More details are given as follows.

##### B. Inputs and Robot Pose Embedding

The inputs include 1) A condition frame  $\mathbf{x}_0^n$  sampled from a video comprising  $N$  frames  $\{\mathbf{x}_0^1, \mathbf{x}_0^2, \dots, \mathbf{x}_0^N\}$ , along with a noisy frame  $\mathbf{x}_t^{n+\Delta k}$  where  $t$  denotes the diffusion steps of  $\mathbf{x}_0^n$ , and  $\Delta k$  represents the frame difference between  $\mathbf{x}_0^n$  and  $\mathbf{x}_0^{n+\Delta k}$ . These frames are concatenated along the channel dimension as the first input. 2) The diffusion time steps  $t$  and frame index difference  $\Delta k$  between the condition frame and the current frame are embedded following Equation (13).

$$\text{emb}(p) = \left( \sin(2^0 \pi p), \cos(2^0 \pi p), \dots, \sin(2^{L-1} \pi p), \cos(2^{L-1} \pi p) \right), \quad (13)$$

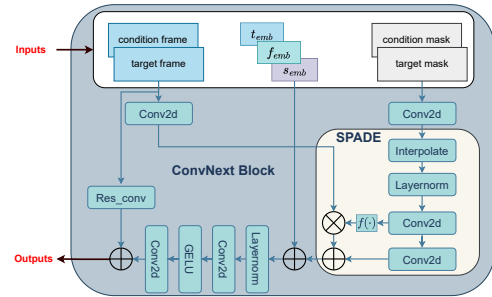


Fig. 3: The improved ConvNext block with a SPADE module. The symbol  $\otimes$  represents element-wise products and  $\oplus$  indicates the sum of two tensors.

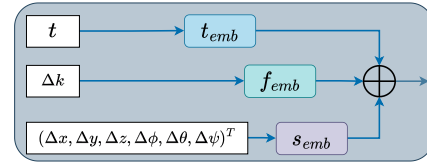


Fig. 4: The embedding block in Fig. 2. We use  $(\Delta x, \Delta y, \Delta z, \Delta \phi, \Delta \theta, \Delta \psi)^T$  to represent the robot pose difference between the condition frame and the current frame (frame to generate), the frame index difference is denoted as  $\Delta k$ , and the diffusion time step is denoted as  $t$ .

where  $p$  represents  $t$  or  $\Delta k$ . We have also embedded the robot pose difference (between the condition frame and the frame to generate) vector  $(\Delta x, \Delta y, \Delta z, \Delta \phi, \Delta \theta, \Delta \psi)^T := \Delta \mathbf{P}$  into each ConvNext block, as shown in Figure 4.  $\Delta x, \Delta y, \Delta z$  are position differences, and  $\Delta \phi, \Delta \theta, \Delta \psi$  are orientation differences, respectively. In this paper, we use a linear embedding strategy for the pose difference vector embedding, i.e.,  $\Delta \mathbf{P}' = \mathbf{A} \cdot \Delta \mathbf{P} + \mathbf{b}$ . The motive behind this is the pose of the robot changes almost linearly as the time between the two frames is short. One setting is  $\mathbf{A}$  is identity and  $\mathbf{b}$  is  $\mathbf{0}$ .

##### C. Mask Regulation

Semantic masks, abundant in shape and location information, have become easily accessible with advancements

in object segmentation models like the Segment Anything model [17]. Recognizing the potential benefits of leveraging semantic mask information, we propose incorporating it into ConvNext blocks to regulate intermediate outputs. Our approach introduces a new SPADE-based ConvNext block, outlined in Figure 3. Initially, frames and masks undergo separate processing through convolutional layers (conv2d), yielding outputs denoted as  $\mathbf{x}$  and  $\mathbf{m}$ , respectively. Subsequently, the output  $\mathbf{m}$  undergoes further processing using the proposed SPADE block to regulate  $\mathbf{x}$ . The SPADE block, as shown in Figure 3, is defined as follows.

$$\bar{\mathbf{x}} = \mathbf{x} \otimes f(\gamma) \oplus \sigma, \quad (14)$$

where the symbol  $\otimes$  indicates element-wise products,  $f(\cdot)$  represents a mapping,  $\bar{\mathbf{x}}$  is the output of the SPADE block,

$$\bar{\mathbf{m}} = \text{Layernorm}\left(\text{Interpolate}(\mathbf{m})\right), \quad (15)$$

$\gamma = \text{conv2d}(\bar{\mathbf{m}})$ , and  $\sigma = \text{conv2d}(\gamma)$ . We use the Layernorm( $\cdot$ ) module to retain information from all mask channels and the nearest neighbor interpolation method is used for the Interpolate( $\cdot$ ) module to ensure the size of masks matches that of the frames.

It is worth mentioning that the SPADE normalization in Equation (14) is different from [6] and [18] as we focus on using mask information to regulate intermediate outputs of ConvNext module such that shape and location information can be retained in video generation.

#### D. Sampling

In the sampling phase, the model is presented with a singular frame extracted from the video to generate subsequent frames interactively. This process continues until the desired number of frames has been produced. During each iteration, the model utilizes the provided frame and conditions such as pose information and semantic masks to inform the generation of the subsequent frames, ensuring a coherent and sequential flow of frames in the generated video.

## V. EXPERIMENTS

### A. Datasets

Given the necessity of robot pose information for training the proposed models, we constructed our datasets accordingly. We employed ROS to control robots in diverse environments, capturing video footage at 24 frames per second (fps). Subsequently, we processed the footage to produce videos with a reduced frame rate of 1 fps, ensuring noticeable changes in the robot’s pose. Our dataset comprises two types of robots: the Turtlebot Waffle Pi robot and a cobot. For the Turtlebot, we recorded videos in two laboratory environments: one with the robot and a simple background (Scene I) and the other with a more complex background (Scene II). Additionally, we recorded the translational and rotational velocities of the robot from ROS to calculate robot pose difference vectors. The frames of these videos were annotated to generate the necessary masks. We also

$\Delta x(m)$	$\Delta y(m)$	$\Delta z(m)$	$\Delta \phi(rad)$	$\Delta \theta(rad)$	$\Delta \psi(rad)$
...	...	...	...	...	...
-1	0	0	0	0	0.251
-0.5	0	0	0	0	0.252
-2	1	0	0	0	0.252
...	...	...	...	...	...

TABLE I: Examples of robot pose data. It is noteworthy that as the robots move on flat floors, there are no translational changes along  $z$  axis ( $\Delta z = 0$ ), and there are no rotational changes along  $x$  ( $\Delta \phi = 0$ ) and  $y$  ( $\Delta \theta = 0$ ) axes. We keep these columns to make the models general to robots work in different environments.

created a third dataset (Scene III) featuring the cobot using a similar procedure to test the adaptability and robustness of our models. It is worth noting that our models focus on retaining the shape and location of objects of interest, such as robots, rather than super-resolution or high-resolution frame generation. Therefore, irrespective of the original frame sizes, we resized both the frames and masks to dimensions  $256 \times 144$  to optimize computational resources and accelerate training. This also facilitates fair comparisons with benchmark models. Our dataset is publicly accessible at: <https://github.com/PengPaulWang/diffusion-robots..>

### B. Models

In this paper, we explore two types of conditions: masks and robot pose information. To comprehensively compare and understand how these different conditions impact the shape and location retention performance of diffusion models, we investigate three models: 1) Ours-Mask-Pose, where both masks and pose information are utilized as conditions; 2) Ours-Mask, where only masks are employed as conditions; and 3) Ours-Pose, where only pose information is used as a condition. SinFusion is employed as the benchmark model for performance evaluation and comparison.

All three of our models utilize a backbone ConvNext consisting of 16 improved blocks, as depicted in Figure 3. To ensure a fair comparison, the benchmark model also employs 16 blocks but lacks pose embedding and mask regulation. Additionally, our models feature several key distinctions: 1) When masks serve as conditions (in Ours-Mask-Pose and Ours-Mask models), they are subjected to regulation via the proposed SPADE module, as illustrated in Figure 3. 2) In instances where robot poses are employed as conditions (in Ours-Mask-Pose and Ours-Pose models), the difference in robot pose between two frames is embedded and integrated into the model, as depicted in Figure 4. Table I presents some example data of the robot pose used for embedding. Notably, these data are collected in ROS, which is trivial.

All models were trained on a single Nvidia A100 GPU. For our models, the loss function in Equation (12) is used for training, while SinFusion training employed Equation (9). Training durations varied among the models: the Ours-Mask-Pose model required approximately 6.2 hours, Ours-Mask took around 5.7 hours, and Ours-Pose took approximately 3.75 hours. In comparison, the benchmark model SinFusion

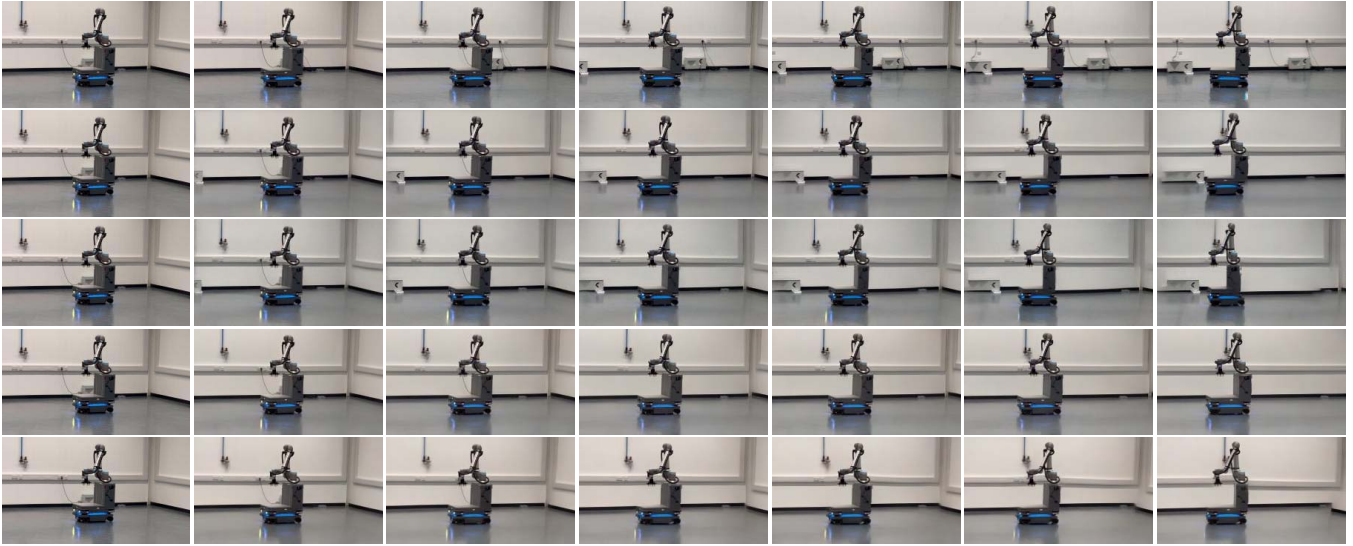


Fig. 5: Results on Scene III, from top to bottom rows: 1) Original frames; 2) Ours-Mask-Pose; 3) Ours-Mask; 4) Ours-Pose; 5) SinFusion. It is noteworthy that only robot masks are used in models where masks are required for this set of results. While robot shape retention can be observed by comparing original and generated frames, location retention can be observed by comparing the robot location with the background in the upper right corner of the frames.

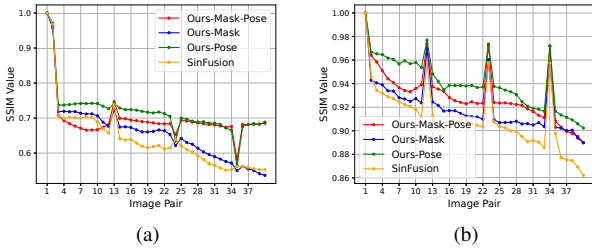


Fig. 6: Quality of generated frames against SSIM. (a) Scene I; (b) Scene III. Note that only robot masks are used.

required approximately 3.72 hours for training.

### C. Evaluation Metrics

Three metrics are employed to assess model performance. SSIM is utilized to evaluate frame generation quality across different models. SSIM is preferred over PSNR for two main reasons: Firstly, SSIM measures image similarity in terms of structural information, luminance, and contrast, providing a more comprehensive assessment compared to PSNR, which solely quantifies reconstruction quality by comparing pixel values between original and generated frames. Secondly, as our focus is on retaining the shape and location of objects in generated frames, SSIM offers a more relevant comparison metric since shape and location information is assessed at a structural level rather than at the pixel level.

Shape-retention performance is evaluated by comparing the Hu moments of the  $i$ -th original frame  $\mathbf{m}_{\text{orig}}^i$  with those of the  $i$ -th generated frame  $\mathbf{m}_{\text{gen}}^i$ . Hu moments are seven real-valued descriptors chosen for their ability to capture essential shape properties of an object of interest. These moments offer a concise representation of shape features,

encompassing characteristics such as orientation, scale, and skewness [19]. Equation (16) is utilized to quantify the shape-retaining performance of diffusion models compared to the original video. The output of Equation (16) indicates the dissimilarity between shapes in the generated frames and their corresponding original frames, with smaller values suggesting greater similarity. More information about Hu moments can be found in the supplemental materials at: <https://github.com/PengPaulWang/diffusion-robots>.

$$d^i = \sqrt{\sum_{j=1}^7 \left( \mathbf{M}_{\text{orig}}^i[j] - \mathbf{M}_{\text{gen}}^i[j] \right)^2}, \quad (16)$$

where  $d^i$  is the similarity between shapes of interest in the  $i$ -th original and generated frames, and  $\mathbf{M}_{\text{orig}}^i[j]$  and  $\mathbf{M}_{\text{gen}}^i[j]$  represent the  $j$ -th Hu moments of the  $i$ -th original and generated frames, respectively.

The Intersection over Union (IoU) metric is utilized to assess the model's performance in retaining the robot's location. Rather than directly determining the precise location of the robot, we employ Equation (17) to calculate the IoU between the masks of the robot in the  $i$ -th original frame  $\mathbf{m}_{\text{orig}}^i$  and the mask of the robot in the  $i$ -th generated frame  $\mathbf{m}_{\text{gen}}^i$ . This computation serves as an indicator of how effectively the location is preserved in the generated videos.

$$\text{IoU}^i = \frac{\mathbf{m}_{\text{orig}}^i \cap \mathbf{m}_{\text{gen}}^i}{\mathbf{m}_{\text{orig}}^i \cup \mathbf{m}_{\text{gen}}^i} \quad (17)$$

### D. Main Results

To delve deeply into the impact of masks and poses on the performance of shape and location retention, we have first considered the masks of the two types of robots exclusively.

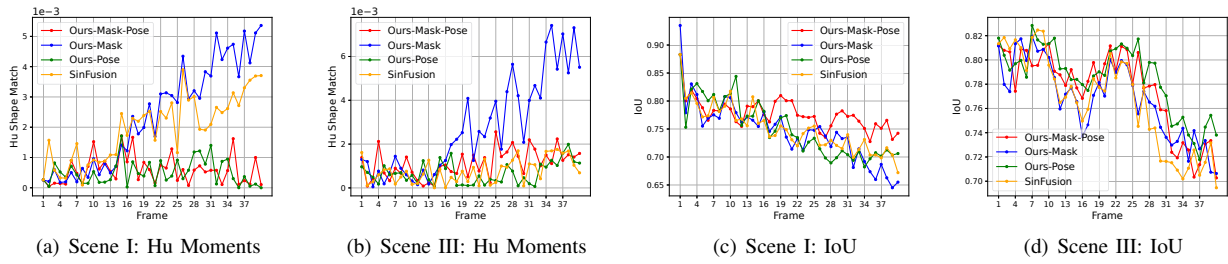


Fig. 7: Shape and location retention performance of different models. Only the robot masks are used where masks are used as conditions for frame generation.

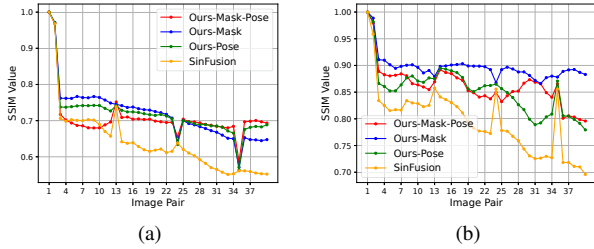


Fig. 8: Quality of generated frames against SSIM. (a) Scene II; (b) Scene III. Both robot and background masks are used.

Two sets of experiments were conducted, one using the Scene I datasets and the other using the Scene III datasets.

The trained models from each dataset are employed to generate frames for evaluation. Figure 5 displays some of the generated results from Scene III, with additional results available in the [https://stummua-cmy.sharepoint.com/personal/55141653\\_a\\_d\\_m\\_mu\\_a\\_c\\_u\\_k/\\_layouts/15/backgroun...](https://stummua-cmy.sharepoint.com/personal/55141653_a_d_m_mu_a_c_u_k/_layouts/15/backgroun...)

Quantitative evaluation results using the three metrics are computed: 1) shape retention based on Equation (16); 2) location retention based on Equation (17); and 3) overall quality of generated frames based on SSIM. The SSIM results are depicted in Figure 6, while the shape and location retention results are illustrated in Figure 7.

Regarding the overall quality of the generated frames, it can be observed from Figure 6 that Ours-Pose achieves the best results, and Ours-Mask-Pose achieves comparable results, but both outperform the benchmark model. Regarding shape and location retention, it is evident from Figure 7

that Ours-Mask-Pose achieves either the best or the second-best results in both aspects. Ours-Pose achieves comparable results with Ours-Mask-Pose in shape retention. In terms of location retention, Ours-Mask-Pose performs comparably with Ours-Pose and outperforms other models in both Scene I and Scene III. In conclusion, incorporating sole pose information or the combination of pose information with masks improves the performance of diffusion models compared to the benchmark model across all three metrics. However, considering only mask results does not always improve the performance compared to the benchmark models, which we assume is due to the exclusive use of robot masks. Further experiments are conducted to investigate this phenomenon.

### E. Ablation Study

1) *Considering Both Robot and Background Masks:* To further investigate the impact of masks on the generation results, additional experiments were conducted on Scene II and Scene III, using masks of both the robots and the backgrounds as conditions. The SSIM results are presented in Figure 8, while the shape and location retention results are depicted in Figure 9. It is evident that by considering both robot and background masks, the quality of generated frames by Ours-Mask has improved in terms of SSIM. In Scene II, Ours-Mask achieves comparable results with Ours-Pose or Ours-Mask-Pose, and in Scene III, it either slightly outperforms Ours-Pose and Ours-Mask-Pose or achieves comparable results. Regarding shape and location retention, improvements are observed with Ours-Mask as well, as shown in Figure 9. However, Ours-Pose and Ours-Mask-Pose still outperform Ours-Mask in both shape and location

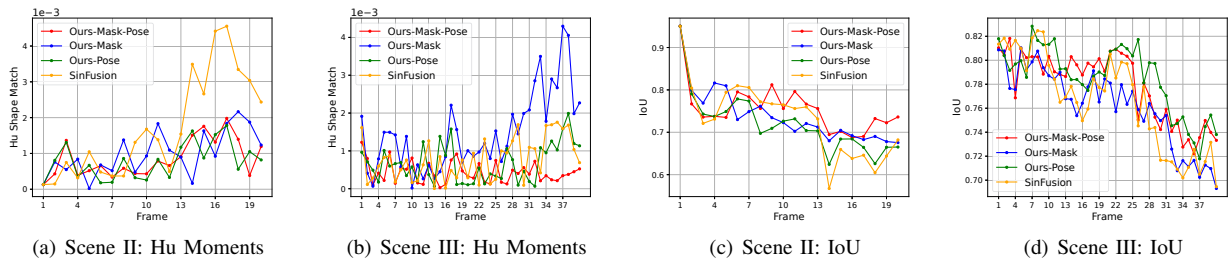


Fig. 9: Shape and location retention performance of different models. Both robot and background masks are used where masks are employed as conditions for frame generation.

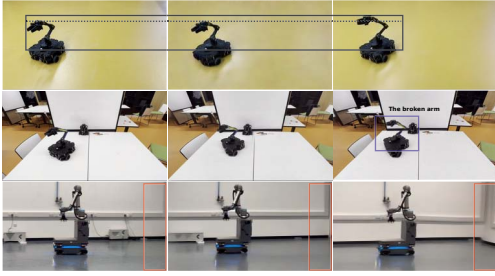


Fig. 10: Evaluation. *Left*: the original frame; *Middle*: Ours-Mask-Pose; *Right*: SinFusion. *Top row*: Ours-Mask-Pose retains the arm shape better; *Middle row*: the robot arm was broken in the frame generated by SinFusion; *Bottom row*: Ours-Mask-Pose retains the location of the robot better.

retention in both scenes.

2) *The implication of Shape and Location Retention*: Some examples of shape and location retention of the robots are provided in Figure 10. In Scene I, Ours-Mask-Pose keeps the shape of the robot better compared to the benchmark model. In Scene II, similar results are observed and the robot arm is broken into two in the generated frame by the benchmark model. The location retention is shown in the results from Scene III, this can be recognized from the relative location of the robot and the wall highlighted.

Considering all experiments across the three scenes, it can be concluded that masks and pose information contribute to retaining the structural information of generated frames. In the meantime, it is important to highlight that models incorporating robot pose embedding only have consistently achieved comparable results in terms of location retention to those incorporating mask regulation, albeit with shorter training times. However, considering robot and/or background masks helps to improve the performance in shape retention and SSIM, but normally needs a longer model training time. Regardless, better performance has been achieved by the proposed models compared to the benchmark model.

## VI. CONCLUSIONS

This paper introduces diffusion models that leverage robot pose and masks as conditional inputs for video generation. The objective is to produce video frames that maintain high structural fidelity, thereby enhancing the preservation of the shape and location information of objects within the generated frames. Through a series of experiments conducted across three distinct scenes involving various robots, we consistently observed improvements in generation quality as measured by SSIM, as well as in the retention of shape and location evaluated using Hu moments and IoU. These advancements hold promise for applications where accurate depiction of robot shape and location is crucial. For instance, our models will be further developed to generate data to facilitate accurate dangerous human-interaction detection training, which will help mitigate potential risks associated with human-robot interactions.

## REFERENCES

- [1] H. Cao, C. Tan, Z. Gao, G. Chen, P.-A. Heng, and S. Z. Li, "A survey on generative diffusion model," *arXiv preprint arXiv:2209.02646*, 2022.
- [2] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [3] T. Höpfe, A. Mehrjou, S. Bauer, D. Nielsen, and A. Dittadi, "Diffusion models for video prediction and infilling," *arXiv preprint arXiv:2206.07696*, 2022.
- [4] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," *arXiv:2204.03458*, 2022.
- [5] T.-S. Chen, C. H. Lin, H.-Y. Tseng, T.-Y. Lin, and M.-H. Yang, "Motion-conditioned diffusion model for controllable video synthesis," *arXiv preprint arXiv:2304.14404*, 2023.
- [6] Y. Nikankin, N. Haim, and M. Irani, "SinFusion: Training diffusion models on a single image or video," *arXiv preprint arXiv:2211.11743*, 2022.
- [7] Y. Okuyama, Y. Endo, and Y. Kanamori, "Diffbody: Diffusion-based pose and shape editing of human images," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 6333–6342.
- [8] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. PMLR, 2015, pp. 2256–2265.
- [9] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
- [10] W. Wang, J. Bao, W. Zhou, D. Chen, D. Chen, L. Yuan, and H. Li, "Semantic image synthesis via diffusion models," *arXiv preprint arXiv:2207.00050*, 2022.
- [11] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.
- [12] O. Ronneberger, P. Fischer, and T. Brox, "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*. Springer, 2015, pp. 234–241.
- [13] R. Yang, P. Srivastava, and S. Mandt, "Diffusion probabilistic modeling for video generation," *arXiv preprint arXiv:2203.09481*, 2022.
- [14] V. Voleti, A. Jolicœur-Martineau, and C. Pal, "Mevd-masked conditional video diffusion for prediction, generation, and interpolation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 371–23 385, 2022.
- [15] S. Wang, J. Zhang, P. Wang, J. Law, R. Calinescu, and L. Mihaylova, "A deep learning-enhanced digital twin framework for improving safety and reliability in human-robot collaborative manufacturing," *Robotics and computer-integrated manufacturing*, vol. 85, p. 102608, 2024.
- [16] K. Holmquist and B. Wandt, "Diffpose: Multi-hypothesis human pose estimation using diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15 977–15 987.
- [17] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.
- [18] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2337–2346.
- [19] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE transactions on information theory*, vol. 8, no. 2, pp. 179–187, 1962.