Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Motion-Aware Conditional Diffusion Framework with Hybrid U-Net and Bidirectional ConvLSTM for Video Frame Prediction

Enver Akbacak

eakbacak@29mayis.edu.tr

Istanbul 29 Mayis University

Nedim Muzoğlu

Sağlık Bilimleri Üniversitesi

Research Article

Keywords: Video frame prediction, Conditional diffusion, ConvLSTM, Unet

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10889712/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Motion-Aware Conditional Diffusion Framework with Hybrid U-Net and Bidirectional ConvLSTM for Video Frame Prediction

Enver Akbacak
29 Mayis University
Faculty of Engineering and Natural Sciences
Department of Electrical and Electronics Engineering
Istanbul/Turkiye
eakbacak@29mayis.edu.tr

Nedim Muzoğlu
University of Health Sciences
Hamidiye Institute of Health Sciences
Department of Bioengineering
Istanbul/Turkiye
nedim.muzoglu@sbu.edu.tr

Abstract

Predicting future video frames remains one of the fundamental challenges in the field of computer vision. The challenge arises from the necessity to capture intricate spatial and temporal dynamics simultaneously. Also known as video prediction, it is a cornerstone of many scientific studies, including autonomous driving, anomaly detection, weather forecasting, and predictive video coding. Current studies either suffer from deterministic blurring caused by pixel-based optimization, error propagation from autoregressive methods, or incur high parameter overhead with unreported computational costs that are hard to implement. To address this challenge, we leveraged the potential of diffusion models, which remain relatively underexplored in the field of video prediction. The framework incorporates a hybrid diffusion-based, bidirectional ConvLSTM architecture within a Unet-style backbone, leveraging the proven efficacy of the Unet in image restoration. Unlike standard practices, this approach combines bidirectional ConvLSTM layers with custom attention mechanisms on Unet to handle temporal continuity, while relying on Denoising Diffusion Probabilistic Models to stabilize spatial details. The proposed framework is suitable for real-world deployment due to its low parameter count and is capable of performing high-fidelity frame prediction. Optimized via a conditioned diffusion process, it effectively uncovers underlying motion information and spatial boundaries, delivering precise future-frame sequence predictions in a single pass while remaining competitive with benchmarks.

Keywords Video frame prediction, Conditional diffusion, ConvLSTM, Unet.

1 Introduction

Future frame prediction is the process of estimating subsequent frames from preceding ones. It is a challenging problem in computer vision that is still the subject of extensive research. Accurate prediction of video states several seconds into the future can provide essential decision-support capabilities for various real-world applications. In autonomous systems, for example, predicting what will be seen in the next few seconds, along with the current situation, and taking action accordingly to plan the safest route is essential. Some studies utilize frame prediction to reduce network-related delays to the millisecond degree in remote robot control [1]. Beyond robotic technology, it offers significant benefits for detecting anomalies in healthcare, such as predicting patient falls before they occur [2]. Videos now dominate the internet. Efficient use of transmission lines is essential in video transmission. If a video server can predict frames a few seconds in advance and the receiving end can do the identical prediction,

then the server can send only the error between the prediction and the actual image, rather than all video frames, thereby significantly reducing congestion on the transmission lines. Therefore, frame prediction is indispensable in predictive video coding. [1], [3], [4], [5].

There are no standard datasets specific to frame prediction. However, datasets prepared for video classification, action recognition, or video understanding are widely utilized. There are only a few studies prepared as comprehensive literature reviews on the topic. One review study emphasizes that determining the cost function is critical [6], and another highlights the importance of model selection [7]. However, our preliminary studies show that both a proper model and an appropriate cost function for corresponding models are critical. Therefore, our literature review will focus primarily on proposed models and suggested cost functions. In the literature, mean squared error (MSE) is the most common cost function, and deep learning-based encoder-decoder models, mostly convolutional LSTM-based, are dominant [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. However, other studies utilize 3D convolutions [19], generative adversarial networks (GANs) for sharper image synthesis [20], and pure Vision Transformers (ViTs) [21], [22], [23], or hybrid models such as transformer-LSTM pipelines [24], Mamda-LSTM [25], CNN-Transformer designs [26], [27], or 3DCNN-LSTM combinations [28], [29]. As summarized in the following, some articles are critiqued immediately after the citation, while others are addressed collectively at the end of the paragraph.

Huang et al. argued that optimizing solely with Mean Squared Error (MSE) loss, even within GAN-based networks, fails to capture complex motion trajectories [30] accurately. They proposed using a motion-oriented network with edge-detection techniques to improve clarity, and replacing the MSE with a composite loss function to capture both motion and content detail better. Convolutional LSTM (ConvLSTM) layers were used in the encoder-decoder architecture to improve motion capture. Upon frame prediction, GANs were used to generate the final clear video frames. Some studies have extended this topic. Li et al. [31] adopted a similar approach to [21], emphasizing the importance of using motion-oriented networks. Ranjan et al. [32] introduced a dual-stream network in which one stream processed frames and the other focused on motion. Both streams used an iterative convolutional encoder-decoder architecture to handle spatial and temporal dependencies efficiently.

Transformer-based models have also attracted attention in video prediction for their ability to capture long-term dependencies. Huang et al. used scalable transformers that process features of different resolutions within a network architecture and employ channel-based attention mechanisms to reduce computational cost. [33]. To focus on motion, they used a dual-stream network similar to that used in [32]. Lee et al. used a convolutional Fourier Inception network to detect long-term temporal relationships. Still, they did not provide conclusive evidence that this approach reduces the need for large training datasets for transformers [5]. Li et al. used a motion-aware Unet architecture for the early detection of patient falls [2]. Mathai et al. integrated LSTMs and Transformers but did not explicitly demonstrate the necessity of combining them, since Transformers are already better at detecting long-range relationships that LSTMs struggle with [5]. Lee et al. argued that CNN-based models are computationally more efficient than RNN and Transformer-based models [34]. They independently processed spatial and temporal data using $1x1$ convolutions and combined

the feature maps to generate spatio-temporal features for future frame prediction; However, they did not assess how much better their experimental results were than those of RNNs and transformers.

There are a few fully generative AI-based (GAN) video prediction models, and their performance is not at the desired level [20]. In addition, GAN-based models often experience imbalances between the generator and discriminator (training instability), or situations that trick the discriminator and produce biased data (mode collapse) when handling complex temporal data. This is not surprising, as GANs are primarily designed to generate artificial, high-quality images.

Although few in number, there are also studies based on the selection of the cost function. Lai et al. proposed using three-dimensional frequency loss instead of MSE. The loss score is obtained in the frequency domain using three-dimensional Fast Fourier Transform (FFT) [35]. This method focused on reducing blurring and on the motion and edge details of the frames using attention mechanisms. However, they did not provide sufficient information on how it affected the temporal flow of frame predictions. Singh et al. used 3D convolutions to encode spatial features of frames while implementing attention mechanisms within a Transformer to process spatiotemporal features [36]. However, the authors did not explain why convolutions were replaced by those feature-based approaches, which are also common in Transformer networks. Mokssit et al. proposed a motion-sensitive model to improve the prediction of frames affected by camera movement [37]. They predicted the possible positions of moving objects using transformer-based models and placed them against a static, unchanging background. However, no information is provided on how they found a solution for changing pixels other than moving objects. The comprehensive literature review reveals that the following critical bottlenecks are commonly observed in recent studies:

The Fidelity-Efficiency Trade-Off: Most of the prominent models either have an excessive number of parameters or lack sufficient empirical validation for practical applications.

Blur Versus Physical Inconsistency: Standard loss formulations often yield blurry averages, whereas purely generative models, such as GANs, can produce structural hallucinations that disrupt physical continuity.

Unjustified Architectural Hybridization: Numerous recent studies combine complex architectural components, such as LSTMs and Transformers, without providing clear ablation studies or theoretical rationale.

The power of diffusion has not been sufficiently utilized in future frame prediction studies. However, Denoising Diffusion Probabilistic Models (DDPMs) transform frame prediction from a passive, pixel-based calculation to an active, observation-based approach [38]. They overcome those fundamental limitations of GANs by learning a probability distribution that eliminates noise, rather than predicting pixels. In this regard, the proposed approach integrates the generative capabilities of denoising diffusion models, the temporal memory retention of ConvLSTMs, and the detail-preserving topology of a U-Net backbone, all coordinated by a localised attention mechanism. The model also uses context frames and noise time steps as conditioning inputs, thereby guiding the denoising process to synthesize temporally coherent, perceptually authentic future frames in a single pass. The highlights of the study are outlined in the following architectural innovations.

Architectural Integration: The Unet-style network, with optimized skip connections, incorporates bidirectional ConvLSTM2D modules positioned at the structural

bottleneck and Denoising Diffusion Models. ConvLSTM2D units condition the iterative diffusion process. Thus, long-term temporal coherence is maintained across predicted frames.

Timestep Conditioned Normalisation: The framework utilises Adaptive Group Normalisation (AdaGN) layers to guide the diffusion trajectory. At each denoising step, the temporal index (timestep embedding) is processed by AdaGN. Thus, the network adaptively modulates feature maps based on the current noise level.

Perceptually Grounded Optimisation: The proposed optimisation technique extends beyond a pure pixel-level loss. An optimised combination of Mean Squared Error (MSE) and Structural Similarity Index Measure (SSIM) loss prioritises structural and perceptual authenticity.

A comprehensive evaluation of the proposed framework was conducted on two benchmark datasets. The empirical results demonstrate that it achieves competitive performance relative to state-of-the-art methods while remaining feasible for practical, real-world deployments given its model size. The remainder of the study is outlined in the following sections. Chapter II details how the model, its architecture, and its sub-components were designed. Chapter III presents key results from detailed experiments, ablation studies, and comparisons with other methods. Finally, Chapter IV is the Conclusion section.

2 The Proposed Method

2.1 Motivation and Problem Formulation

The objective of video frame prediction is to synthesize future video frames by learning spatio-temporal dynamics from known context frames. Let a video frame sequence be expressed as a tensor of the form $\chi = \{x_1, x_2, \dots x_T\}$, where any frame size is $\mathcal{R}^{NxNxD}$, D represents the channel dimension, and NxN is the size of any frame. Suppose M consecutive historical context frames are presented as follows,

$$\chi_{context} = \{x_{t-M}, x_{t-M+1}, \dots, x_{t-1}\} \quad (1)$$

Depending on the target temporal horizon, the task is divided into two operational scenarios: Single-Frame Prediction (SFP) and Multi-Frame Prediction (MFP). In the SFP scenario, the network is parameterised to directly project the past context frames to the next target frame x_T. In the conditional generative diffusion perspective, the target frame is regarded as a latent state vector, x_t^τ, corrupted by noise at a specific diffusion time step $\tau \sim \cup(0, \mathcal{T} - 1)$. Consequently, the model $\mathcal{M}_\vartheta$ is optimised to map the combined sequence of context frames and the noisy target frame to a sharp reconstruction of the clean target frame in the training phase as follows,

$$\hat{x}_t = \{\mathcal{M}_\vartheta[\, x_{t-M}, x_{t-M+1}, \dots, x_{t-1}, x_t^\tau \,], \tau\} \quad (2)$$

During inference, the target frame is completely hidden from the model. The model predicts a future frame, which is at a maximum noise level ($\tau = \mathcal{T} - 1$), in a single denoising step, as follows,

$$\hat{x}_t = \{\mathcal{M}_\vartheta[\, x_{t-M}, x_{t-M+1}, \dots, x_{t-1}, \epsilon], \tau = \mathcal{T} - 1\}, \; \epsilon_K \sim \mathcal{N}(0, \boldsymbol{I}) \quad (3)$$

where ϵ represents the standard Gaussian noise with zero mean and identity matrix variance.

In the MFP scenario, the model simultaneously predicts a sequence of targets, $\hat{X}_t = \{x_t, x_{t+1}, \dots, x_{t+K-1}\}$, consisting of K future frames. The model fulfills the sequence prediction in a single pass, rather than using standard error-accumulating autoregressive (autoregressive roll-out) strategies. During training, the target sequence is degraded according to the

diffusion noise schedule and transformed into a random noise-injected sequence $\{x_t^\tau, x_{t+1}^\tau, \ldots x_{t+K-1}^\tau\}$. The optimization process is then stated as follows,

$$\{\hat{x}_t, \ldots, \hat{x}_{t+K-1}\} = \mathcal{M}_\vartheta([\, x_{t-M}, \ldots, x_{t-1}, x_t^\tau, \ldots, x_{t+K-1}^\tau \,], \tau) \quad (4)$$

This process creates an input tensor with $\mathcal{R}^{Bx(M+K)xNxNxD)}$ dimensions. The inference mechanism eliminates accumulation of errors (autoregressive) from step-by-step prediction, a chronic problem in next-frame prediction literature. However, the proposed method uses purely isotropic Gaussian noise vectors, along with context frames, to generate the next-frame sequence in a single step as follows,

$$\{\hat{x}_t, \ldots, \hat{x}_{t+K-1}\} = \mathcal{M}_\vartheta([\, x_{t-M}, \ldots, x_{t-1}, \epsilon_1, \ldots, \epsilon_K \,], \tau = \mathcal{T} - 1)). \quad (5)$$

The non-autoregressive conditional structure of the developed architecture eliminates the exposure bias inherent in traditional teacher-forcing approaches and the error accumulation problems seen in autoregressive strategies. As a result, the model performs with high spatial accuracy and perfect temporal continuity in both SFP and MFP scenarios.

2.2 Architecture Description

The proposed architecture relies on a hierarchical Unet-style network to capture multi-scale spatio-temporal features. Within the encoder block, ConvLSTM2D layers extract spatio-temporal features simultaneously. As the forward prediction time increases, the frames generated by pure ConvLSTM layers become blurry due to the averaging of probabilities. The power of diffusion comes into play for sharpening and detail generation. Thanks to optimized skip connections, fine-grained and subtle edge details are transferred directly to the decoder block.

To condition the model on the diffusion process, the extracted latent features are modulated by AdaGN layers according to the specific diffusion noise level. Unlike standard autoregressive models, it uses noise-injected ground-truth frames, along with context (historical) frames, as the input sequence during training. In this respect, it differs from teacher-forcing-based methods [39]. During inference, it uses pure noise along with context frames as an input sequence. Using the diffusion technique, the model gradually reduces noise, thereby learning to produce high-quality, sharp feature frames with reduced blur.

Figure 1 depicts the general structure of the proposed model for both SFP and MFP. However, the difference in their implementation lies solely in the management and optimization of temporal dimensions. The operating principles of both cases are explained below, with the differences in mind.

2.3 Input Format Types and Data Flow

MFP predicts multiple frames at once. It considers pure context frames and noise-injected ground truth frames for the diffusion process as an input tensor of size $R^{Bx\,(M+K)xNxNxD}$, where B represents the batch size, M is the number of context frames, and K is the number of frames to be predicted. As stated before, N is the size of frames, and D is the channel dimension. It is particularly important to emphasize that noise-injected ground truths are used with context frames only during the training phase. In the testing phase, pure Gaussian noise is used as input along with context frames. SFP, on the other hand, consists of a ground truth frame with Gaussian noise added, along with clean context frames. Therefore, the input tensor is of size $\mathcal{R}^{Bx(M+1)xNxNxD}$.

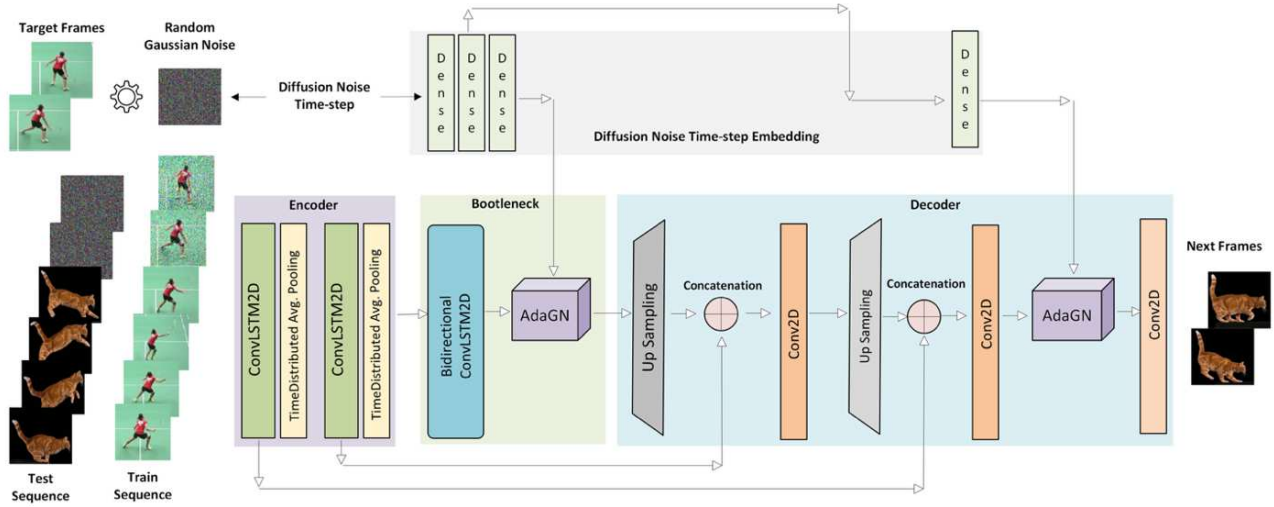


Fig. 1 An overview of the proposed architecture. The input context frames are processed by an Encoder using ConvLSTM2D and TimeDistributed Average Pooling layers, while target slots are initialized with noise based on the sampled diffusion timestep. To enforce temporal conditioning, the noise timestep is projected into a vector embedding and dynamically integrated with the bottleneck's Bidirectional ConvLSTM2D features via Adaptive Group Normalization (AdaGN) layers. The Decoder pathway progressively reconstructs the spatial resolution using UpSampling and Conv2D layers, incorporating skip-connections from the encoder to preserve fine-grained spatial boundaries. Consequently, the proposed pipeline synthesizes high-fidelity next-frame predictions in a single forward pass without requiring iterative sampling loops or teacher forcing.

2.4 Spatio-Temporal Handling Strategies

In MFP, the bidirectional ConvLSTMs in the bottleneck preserve the temporal dimension (return_sequence = True). *Accordingly, in the decoder, all CNN2D and upsampling layers are wrapped with TimeDistributed layers to preserve the temporal dimension.* Therefore, skip connections from the encoder are also connected to the decoder, encompassing all temporal dimensions.

In SFP, the bidirectional convLSTM2D layer at the bottleneck terminates the temporal dimension by scanning the motion history of the input frame sequence, reducing it to a static 2D feature map. Therefore, TimeDistributed layers are not used in the decoder.

2.5 AdaGN and Outputs

Both models use a three-layer neural network (NN) to map the diffusion noise *timestep* variable τ to an embedding vector, which is then used as a conditioning signal in AdaGN. MFP applies time embedding to the input tensor via TimeDistributed AdaGN using repeat-vector parameters. SFP, on the other hand, modulates a single static feature map using a time embedding directly within the 2D AdaGN layers.

2.6 Diffusion and AdaGN Collaboration

Diffusion models can be thought of as an experiment involving mixing ink in one place and then cleaning it up. During the training stage, the model should be drilled for the entire noise spectrum. Therefore, τ is randomly sampled from a uniform distribution. A low τ triggers low noise injection, while a high τ triggers high noise injection. In the validation phase, the model weights are not updated; only the noise removal capability is tested. In this phase, the τ is also randomly sampled from a uniform distribution. In the inference phase, no ground-truth frame is provided, but pure Gaussian noise. Thus, the trained model generates a deterministic reconstruction from pure noise.

In diffusion-based models, noise is gradually injected into pure frames. The τ triggers two

additional variables: one controls the presence of the original pixels, and the other controls the ratio of Gaussian noise to be added. To be more specific, $\boldsymbol{\beta}$ controls the level of pure noise to be injected. It is sampled from the cosine function. At low levels, the noise deviation is minimal; at medium values, it increases, then slows again towards the end. $\boldsymbol{\alpha_\tau}$ indicates the percentage of original pixel presence in a frame at a τ level. It is obtained by multiplying all the α values from no noise level to the level τ as follows,

$$\alpha_\tau = \prod_{i=1}^{\tau}(1-\beta_i). \quad (6)$$

Here, a notable property of the Gaussian distribution comes into play. The product of two Gaussian distributions is still a Gaussian. Therefore, noise injection processes can be performed in a single step, eliminating intermediate stages. Thus, α_τ is precisely the product of the Gaussians. A noisy frame is synthesised in a single step as follows.

$$x^\tau = \sqrt{\alpha_\tau}x_0 + \sqrt{(1-\alpha_\tau)}\,\epsilon \quad (7)$$

where x_0 represents a pure target frame, and ϵ represents the standard Gaussian noise with zero mean and identity matrix variance,
$\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$.

While standard group normalization techniques fix the features, the AdaGN dynamically modulates the filters at each diffusion level τ. The integer value τ is then transformed into a 512-dimensional embedding vector using a three-layer NN with Swish activation, making it suitable for network processing, as

$$\tau_{emb} = Swish(Dense(Swish(Dense(\tau)))).$$

As input to the AdaGN layer, τ_{emb} is split by a linear layer into the scale γ and shift b parameters that guide the normalization process. The mathematical formulation of AdaGN is as follows,

$$AdaGN(x, \tau_{emb}) = GN(x)\,.\,\big(1 + \gamma(\tau_{emb})\big) + b(\tau_{emb}) \quad (8)$$

During the training phase, the model is also expected to learn all noise spectra. Since τ is randomly sampled from a uniform distribution, AdaGN layers should also dynamically process layer features according to the noise level. At high τ values, the AdaGN parameters (γ, b) modulate the convolutional filters to focus on semantic features (global context), since frames are noisy and fine details are indistinguishable. At low τ values, however, they modulate the filters to focus on fine details, namely edges and textures.

In the inference phase, the target frame is completely unknown, so the strategy shifts to a deterministic noise-reduction step. That is, τ is set to the highest noise scale, applying pure Gaussian noise to the model. When the model is fed pure Gaussian noise along with clean context frames, the maximum τ is simultaneously transmitted to the AdaGN layers via a skip connection. In this sense, the model analyzes context frames without investigating fine-grained details, performing a clean frame prediction from pure Gaussian noise in a single step.

2.7 Loss Function and Optimization Objective

The model training is structured to simultaneously optimize pixel-level accuracy and the high-frequency structural quality of the predicted frames. The standard Mean Squared Error (MSE) loss, commonly used in conventional video prediction models, tends to produce deterministic averaging, leading to blurring artifacts at the pixel level. To prevent those artifacts, a hybrid optimization objective has been formulated that combines the MSE loss, which penalizes pixel-level density differences, with the Structural Similarity Index ($SSIM$) loss, which preserves edge and texture integrity, thereby being suitable for human visual perception. In this context, the total optimization loss between the predicted frame sequence and the ground truth sequence is defined as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{MSE}(x_t, \hat{x}_t) + \lambda.\mathcal{L}_{SSIM}(x_t, \hat{x}_t) \quad (9)$$

where x_t and $\hat{x}_t$ represent the predicted and ground-truth frame sequences, respectively, while λ is a hyperparameter weighting coefficient, set experimentally to 2.0, that balances spatial sharpness and structural fidelity.

The first component, $\mathcal{L}_{MSE}$, computes the average squared discrepancy between the predicted frame and the ground-truth frame across spatial pixels, enabling the model to approximate the overall color and intensity distributions:

$$\mathcal{L}_{MSE}(x_t, \hat{x}_t) = \frac{1}{N \times N \times D} \sum_{i=1}^{N} \sum_{J=1}^{N} \sum_{c=1}^{D} (x_t(i,j,c) - \hat{x}_t(i,j,c))^2 \quad (10)$$

The second component is an $SSIM$-based loss designed to preserve the clarity, contrast, and object boundaries of the predicted frames as perceived by the human eye. Due to the hyperbolic tangent activation function ($tanh$) used in the last layer of the network, the pixel density values are within the dynamic range of [-1, 1]. For the standard $SSIM$ function to operate stably in the range of [0, 1], the inputs are normalized to the [0, 1] scale using the $\frac{x+1}{2}$ transformation, as shown in the formula,

$$\mathcal{L}_{SSIM}(x_t, \hat{x}_t) = 1.0 - \text{SSIM}\left(\frac{x_t+1}{2} - \frac{\hat{x}_t+1}{2}\right). \quad (11)$$

Since the standard $SSIM$ index reaches a maximum of 1.0 when two frames are identical, the function has been reversed to $1.0 - \text{SSIM}$ so that the optimization proceeds in a minimization direction. SSIM, on the other hand, is given as follows,

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (12)$$

where μ_x and μ_y are the local means of the pixel densities in each frame; these parameters are used to compare brightness at the level the human eye can distinguish. σ_x^2 and σ_y^2 are the local variances of each frame, while σ_{xy} is the local covariance. These terms evaluate the similarity of the image's contrast properties by measuring pixel signal amplitudes. Examining the linear dependence between the two frames helps preserve the structural integrity of objects, such as edges, textures, and forms. C_1 and C_2 are constants added to the formula to ensure mathematical stability. They prevent *division by zero*, especially in flat or dark background regions where local means or variances approach zero.

Thanks to this formulation, the model not only penalizes pixel-level errors in the diffusion-based noise-reduction process but also produces high-fidelity outputs by optimizing for structural deformations and boundary ambiguities along the object's path of motion during prediction.

3 Experimental Results

The performance of the non-autoregressive conditional diffusion model is evaluated on two benchmark datasets. Comprehensive experiments demonstrate the model's ability to preserve spatial clarity while maintaining temporal consistency. Moreover, the model's low computational cost makes it suitable for real-time applications. The proposed model has only 2.226 million parameters with a size of 8.49 MB on disk. We also note that classical diffusion models are time-consuming to predict frames since a loop is operated during inference. The proposed model, whether SFP or MFP, is ideal for real-time applications as it performs frame predictions in a single step.

3.1 Experimental Setup and Hyperparameters

To ensure the rigorous verification and generalization of the framework, datasets were partitioned into training, validation, and independent testing subsets in an 80:10:10 ratio to prevent data leakage and exposure bias during model evaluation. Video frames from the MNIST3D and KTI datasets were spatially resized to $64x64x1$ and $128x128x3$,

respectively, and normalized to the $[-1.0, 1.0]$ dynamic range to align with the noise distribution of the diffusion model.

The architecture was built using $TensorFlow\ 2.x$ and $Keras\ core$, utilizing Keras Ops for tensor operations. Optimization employed the Adam optimizer with an initial learning rate of $1x10^{-4}$. To maintain training stability and prevent gradient issues in the UNet backbone, an automatic learning-rate decay was implemented. If validation SSIM did not improve for 5 consecutive epochs, the learning rate was halved ($factor = 0.5$). The batch size was set at 16 for up to 100 epochs. For the diffusion process, the maximum time horizon experimentally was set to $\mathcal{T} = 200$, and during training, the diffusion timestep (τ) was sampled uniformly from $\tau \sim \cup(0, \mathcal{T} - 1)$.

3.2 Evaluation Metrics

The accuracy of predictions and the clarity of the structure in the generated future video frames were quantitatively evaluated using two widely adopted benchmark metrics in video processing. MSE measures the average squared difference in intensity between the predicted and target frames. Naturally, a low MSE value indicates higher-quality frame prediction.

3.2.1 PSNR (Peak Signal-to-Noise Ratio)

This metric assesses pixel-level reconstruction accuracy and evaluates absolute intensity deviations. A higher $PSNR$ value signifies that the model effectively mitigates noise and delivers superior pixel-level fidelity to the original. $PSNR$ is expressed on a logarithmic scale in decibels (dB), and defined as,

$$PSNR = 10 \log_{10} \left(\frac{L^2}{MSE} \right) \quad (13)$$

where L represents the dynamic range of the pixel intensities in an image, and MSE is the mean squared error. As seen, MSE is inversely proportional to $PSNR$. As MSE approaches zero, $PSNR$ approaches infinity. Therefore, a higher $PSNR$ value indicates the superiority level of the model.

3.2.2 SSIM (Structural Similarity Index)

It models the human visual perception system (HVS) to evaluate brightness, contrast, and structural boundaries within local spatial windows. High $SSIM$ values indicate that moving objects can be preserved with sharp edges without being subjected to deterministic blurring.

3.3 Datasets

The MNIST3D [40] dataset is a three-dimensional representation of the classic two-dimensional MNIST handwritten digits dataset and is widely used in the machine learning literature. This dataset, containing numbers from 0 to 9, serves as a lightweight, effective tool for testing and benchmarking 3D computer vision, 3D convolutional neural networks (3D CNNs), and point cloud classification algorithms. The videos were recorded in grayscale at a resolution of 64x64 pixels.

The KTH [41] is one of the fundamental datasets in computer vision and video processing literature for evaluating action recognition and video prediction algorithms. The dataset comprises six basic human movements (walking, jogging, running, boxing, hand waving, and hand clapping) performed by 25 individuals. The filming was done against a fixed background under 4 scenarios: an open area, different scales in the open area, different costumes, and an indoor setting. The videos were recorded in grayscale at a resolution of 160 x 120 pixels and 25 frames per second (fps).

In the experiments, 20 frames were extracted from each video, and the first 10 were utilized as context frames. For SPF, only the 11th frame is the target frame, while for the MFP, frames 11-20 are the target frames as a whole. Mnist3D dataset frames are resized to 64x64, while KTI frames are resized to 128x128, as in similar studies for the baseline comparison.

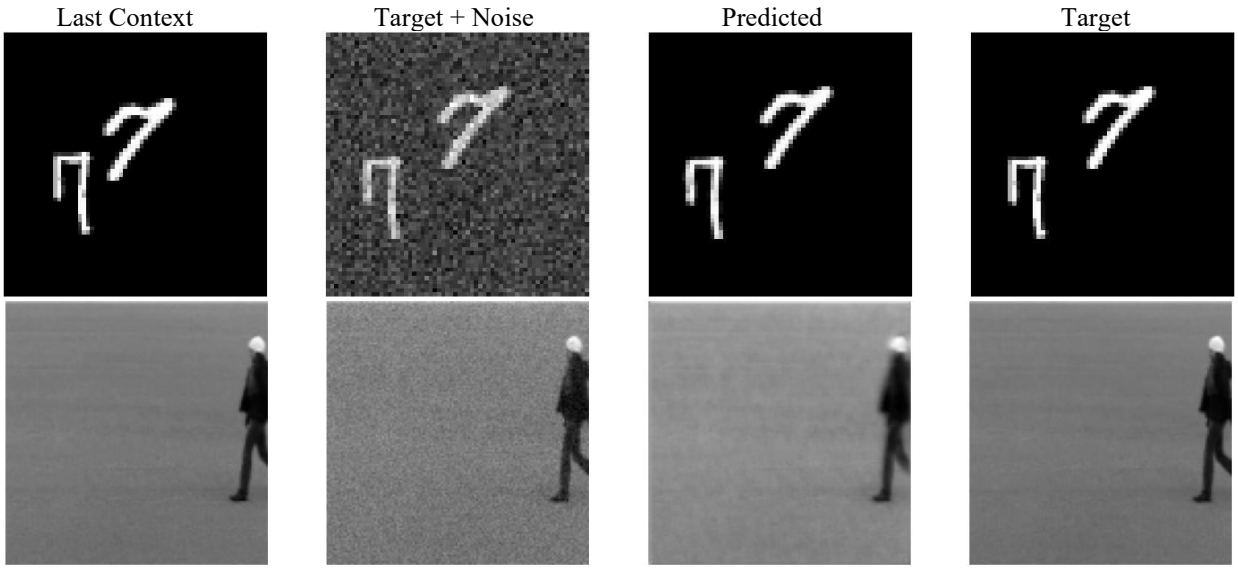


Fig. 2 The leftmost frame is the last observed context frame, followed by the target frame mixed with random Gaussian noise. The model predicts the third frame, and the far-right frame is the target frame.

3.4 Qualitative Analysis

3.4.1 SFP

To validate the proposed model's generative capabilities and spatio-temporal consistency, visual outputs from the training, validation, and testing phases were examined. During the testing phase, the model was supplied solely with clean historical context frames, while pure Gaussian noise was employed to substitute future frames.

For the SFP, frame transitions are shown in Figure 2, based on randomly selected samples. As you can see, the noise has been successfully cleaned up, and the next frame is visually quite close to the target frame.

While traditional deep learning models often yield indistinct results due to error amplification in the presence of pure noise, the proposed architecture produces sharper, more precise outputs. The bidirectional ConvLSTM2D layers at the bottleneck capture robust spatiotemporal memory by analyzing historical motion data; the AdaGN layers dynamically adjust the convolutional filters based on the noise level. In essence, the model can transform random noise into geometrically accurate, well-defined numerical shapes in a single forward pass, without the need for computationally intensive repeated reverse diffusion processes.

3.4.2 MFP

SFP demonstrates absolute spatial superiority thanks to its focused single-frame target horizon. This is because predicting a single frame is always an easier optimization goal than predicting 5-10 future frames simultaneously. Indeed, thanks to the hybrid optimization strategy used and the extraordinary effect that diffusion adds to model training, it has successfully suppressed the chronic blurring tendencies seen in classical models. The boundary and edge details of moving objects have been preserved, and despite abrupt changes in trajectory, their morphological geometry remains intact. Figure 3-4 shows predicted frames alongside the context and ground-truth frames.

As seen in Figure 3, the approach successfully captures motion and produces

sharp, temporally coherent frames. Similarly, the model accurately tracks a human action,

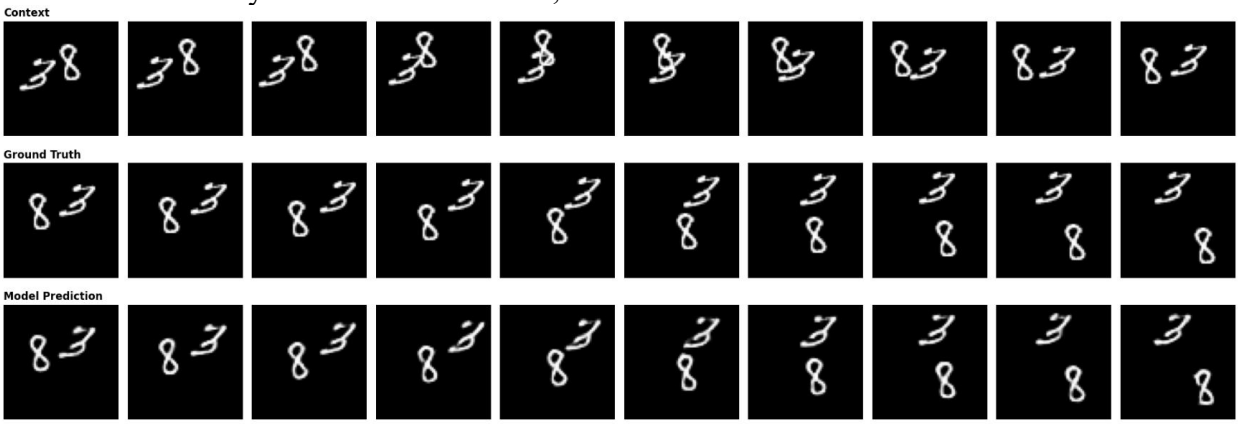


Fig. 3 A qualitative frame prediction experiment taken from MNIST3D. The top row shows the historical context sequence of 10 frames, while the middle row depicts the corresponding ground-truth frames. The bottom row showcases the multi-frame predictions generated by the proposed model with a single forward pass.

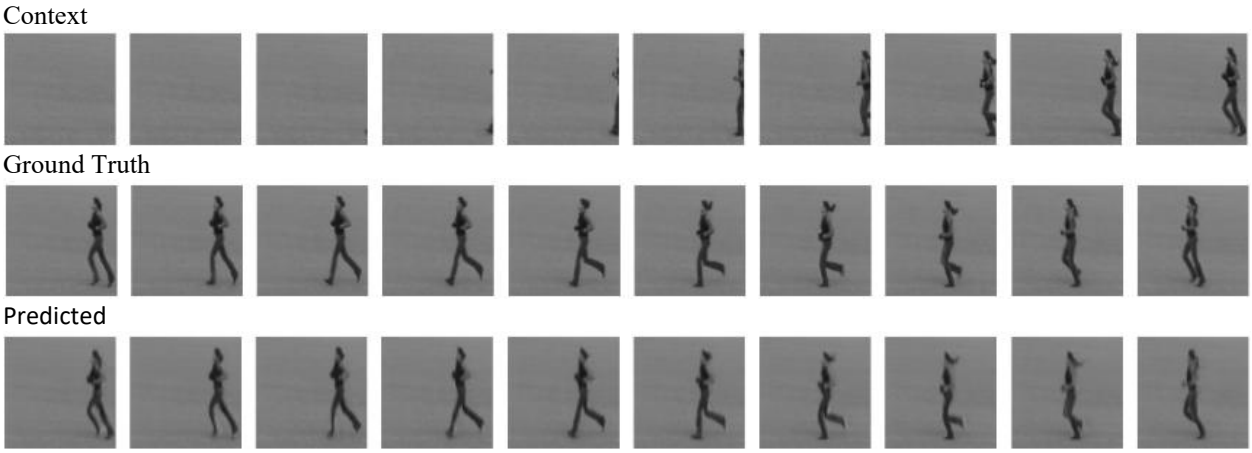


Fig. 4 A frame prediction experiment from the KTI dataset. The consecutive frames depict a person entering the scene and running against a static background. The top row shows the historical context sequence of 10 frames, while the middle row depicts the corresponding ground-truth frames. The bottom row showcases the multi-frame predictions generated by the proposed model by a single forward pass.

Table 1 The proposed method's PSNR and SSIM scores on Mnist3D and KTH datasets.

	MNIST3D		**KTH**	
	PSNR	SSIM	PSNR	SSIM
SFP	27.54	0.962	22.16	0.931
MFP	22.03	0.928	19.03	0.897

as seen in Figure 4. The visual outputs confirm that conditional diffusion effectively suppresses blurring.

The evaluations on the MNIST3D and KTH datasets are summarized in Table 1 for both SFP and MFP, in terms of PSNR and SSIM. The SFP scores highlight the

efficiency of collapsing the temporal axis at the UNet bottleneck and then directly reconstructing the next future frame. The

more challenging task, MFP, on the other hand, still maintains strong SSIM scores.

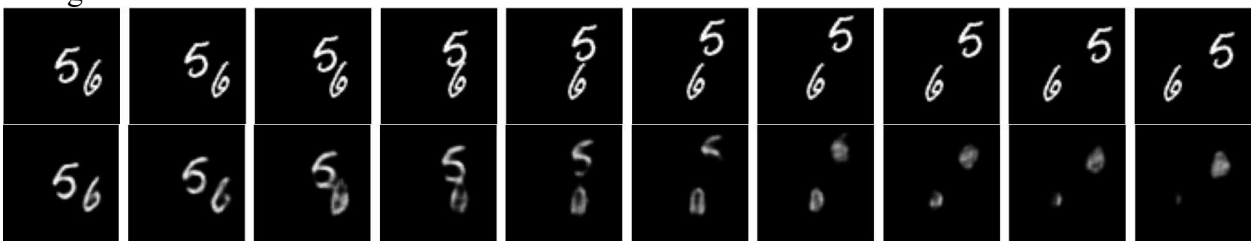


Fig. 5 The effect of the absence of diffusion in MFP. The upper line is the ground-truth frames, and the bottom line is the predicted frames.



Fig. 6 The effect of the absence of diffusion in SFP. From left to right, the last context frame, predicted frame, and ground truth frame.

3.5 Ablation Study

Two control variants were tested to validate the specific contributions of the components in the hybrid architecture.

3.5.1 Effect of Diffusion Conditioning

When the model is stripped of the diffusion process, a significant degradation in spatial coherence is observed. Although the model accurately detects the general direction of movement of the objects, typical blurring and error accumulation occur, especially along the boundaries of the objects. A typical output is presented in Figures 5-6.

3.5.2 ***In the absence of the ConvLSTM Bottleneck*** When temporal memory layers and LSTMs are excluded, and features are directly input into standard 2D convolutions, the model is unable to track moving objects. In addition, serious spatial distortions are observed. These findings highlight that ConvLSTM2D blocks are essential for maintaining temporal continuity and tracking nonlinear motion. We also run experiments by replacing ConvLSTMs with Conv3D. 3D convolutions increase computational costs while decreasing SSIM scores. In both experiments, the SSIM scores decreased noticeably, so no table is presented for the relevant scores. As shown in Figure 7, the absence of the ConvLSTM layers significantly degrades the visual outputs.

3.6 Quantitative Comparisons

Papers that used the datasets employed in our experiments were included in the comparative analyses. Table 1 presents metrics from the independent test set and performance comparisons with other methods. The quantitative results are taken directly from the respective papers. Although various papers do not mention this, in our test phase experiments our model was not exposed to any true ground truth. The ground-truth frames were initialized entirely with pure isotropic Gaussian noise ($\tau = \mathcal{T} - 1 = 199$). In a video with an FPS of 30, SFP may not be necessary in most applications because the next frame is shown every 1/30 of a second. However, SFP also indirectly affects the success of MFP. Since almost all studies in the literature present experimental values in terms of MFP, quantitative comparisons are conducted in terms of MFP by predicting 10 future frames from 10 context frames.

Tables 2-3 outline MFP benchmarking results on MNIST3D and KTH datasets. One of the most important aspects of our model is its suitability for real-time systems, with

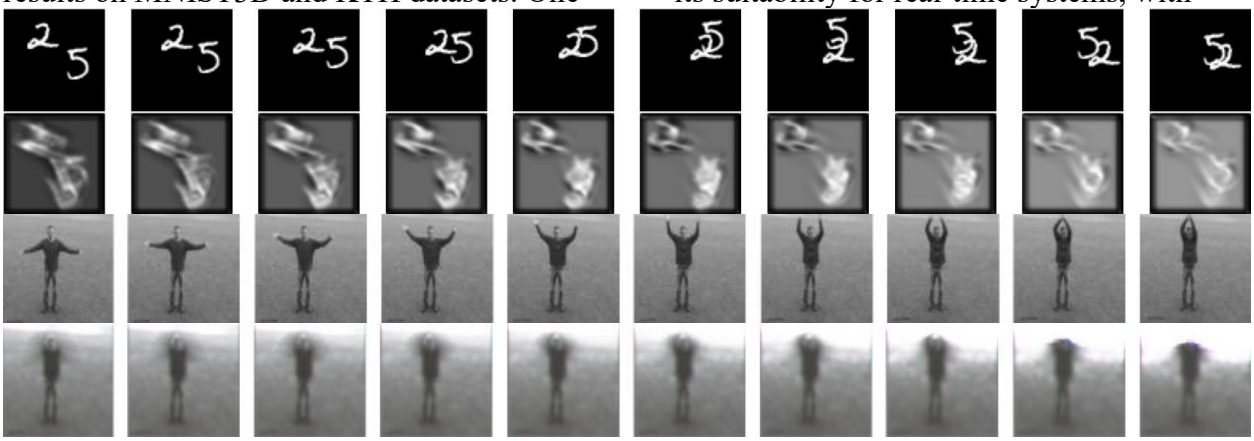


Fig. 7 The effect of the absence of the ConvLSTM layers in the bottleneck. Top lines are ground truths, and bottom lines are predicted frames.

Table 2 Quantitative MFP performance on MNIST3D dataset

Architecture	PSNR	SSIM	Model Size (MB)	Parameters (M)
PredRNN++'18 **[16]**	-	0.898	-	-
E3D-LSTM'19 **[29]**	-	0.910	-	-
Conv-TT-LSTM'20 **[17]**	-	0.915	-	2.69
PredRNN'21 **[15]**	-	0.891	23.86	-
LMC-M'21 **[18]**	-	0.924	-	-
VPTR'22 **[23]**	-	0.844	-	-
SPDE-VP'24 **[10]**	23.91	0.924	-	-
ConvLSTM-3DCNN'25 **[28]**	**28.95**	0.890	-	-
3DTransLSTM'25 **[42]**	23.57	0.921	56.6	11.6
HRS-AE-EFAR'26 **[26]**	-	**0.961**	-	-
Proposed	22.03	0.928	**8.49**	**2.226**

Table 3 Quantitative frame prediction performance on KTI dataset

Architecture	PSNR	SSIM	Model Size (MB)	Parameters (M)
PredRNN++'18 **[16]**	28.47	0.865	-	-
E3D-LSTM'19 **[29]**	29.31	0.879	-	-
PredRNN'21 **[15]**	28.37	0.839	23.86	-
MAFE'21 **[11]**	23.74	0.749	-	-
VPTR'22 **[23]**	26.13	0.859	-	-
MCnet'22 **[30]**	27.37	0.836	-	-
SPDE'24 **[10]**	27.97	0.887	-	-
VMRNN'24 **[25]**	30.06	**0.907**	-	11.6
3DTransLSTM'24 **[42]**	28.78	0.890	-	-
ML3DVP'25 **[19]**	**33.43**	0.794	-	-
HRS-AE-EFAR'26 **[26]**	30.27	0.868	-	-
Proposed	19.03	0.897	**8.49**	**2.226**

2.226 million parameters and 8.49MB of disk space. Many studies have not provided sufficient data on this matter. However, compared with studies that provide parameter estimates, our model performs best. Given the low number of parameters, our model is

significantly better than the majority of studies and has competitive scores. A high

PSNR score alone does not indicate that semantic integrity is maintained and visually coherent prediction is achieved. Even though the model's PSNR score is slightly lower than other models', the most important score, SSIM, which indicates visual accuracy, is quite good given its low number of parameters. As a result, object edges are visually clear and free of blurring.

4 Conclusion

A diffusion-based, bidirectional ConvLSTM architecture within a Unet-style model is proposed to address one of the challenging problems in computer vision: future frame prediction. The model is suitable for real-world deployments due to its low number of parameters and is capable of achieving high-fidelity frame prediction. The proposed approach eliminates deterministic blurring caused by pixel-based models and frame degradation, which is frequently seen in autoregressive methods due to error accumulation. The developed architecture ensures temporal continuity via Bidirectional ConvLSTM layers and custom attention mechanisms within a proven UNet backbone. Most importantly, it leverages the power of conditional diffusion to ensure spatial coherence and stabilize object boundaries in a single forward pass, without using iterative prediction cycles as in autoregressive models. Experiments conducted on two benchmark datasets confirmed that the model, despite its low number of parameters and minimal computational load, exhibits strong competition with existing benchmarks.

Funding Declaration

The authors declare that this research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

[1] T. Sato *et al.*, "Compensation of Communication Latency in Remote Monitoring Systems by Video Prediction," *IEICE Transactions on Communications*, vol. E107.B, no. 12, pp. 945–954, Dec. 2024, doi: 10.23919/transcom.2024CEP0004.

[2] S. Li and X. Song, "Future Frame Prediction Network for Human Fall Detection in Surveillance Videos," *IEEE Sens. J.*, vol. 23, no. 13, pp. 14460–14470, Jul. 2023, doi: 10.1109/JSEN.2023.3276891.

[3] P. Li, C. Zhang, and X. Xu, "Fast Fourier Inception Networks for Occluded Video Prediction," *IEEE Trans. Multimedia*, vol. 26, pp. 3418–3429, 2024, doi: 10.1109/TMM.2023.3310330.

[4] Z. Straka, T. Svoboda, and M. Hoffmann, "PreCNet: Next-Frame Video Prediction Based on Predictive Coding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 8, pp. 10353–10367, 2024, doi: 10.1109/TNNLS.2023.3240857.

[5] J. K. Lee, N. Kim, S. Cho, and J. W. Kang, "Deep video prediction network-ased inter-frame coding in HEVC," *IEEE Access*, vol. 8, pp. 95906–95917, 2020, doi: 10.1109/ACCESS.2020.2993566.

[6] S. Oprea *et al.*, "A Review on Deep Learning Techniques for Video Prediction," Jun. 01, 2022, *IEEE Computer Society*. doi: 10.1109/TPAMI.2020.3045007.

[7] Y. Zhou, H. Dong, and A. El Saddik, "Deep Learning in Next-Frame Prediction: A Benchmark Review," 2020, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2020.2987281.

[8] Y. Du, J. Zhang, X. Cheng, Y. Lu, D. Li, and W. Tian, "Predicting Sudden Stratospheric Warmings Using Video Prediction Methods," *Geophys. Res. Lett.*, vol. 52, no. 8, Apr. 2025, doi:

10.1029/2024GL113993.
[9] N. Kim and J. W. Kang, “Dynamic motion estimation and evolution video prediction network,” *IEEE Trans. Multimedia*, vol. 23, pp. 3986–3998, 2021, doi: 10.1109/TMM.2020.3035281.
[10] X. Wu, J. Lu, Z. Yan, and G. Zhang, “Disentangling Stochastic PDE Dynamics for Unsupervised Video Prediction,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 15427–15441, 2024, doi: 10.1109/TNNLS.2023.3286890.
[11] X. Lin, Q. Zou, X. Xu, Y. Huang, and Y. Tian, “Motion-Aware Feature Enhancement Network for Video Prediction,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 2, pp. 688–700, Feb. 2021, doi: 10.1109/TCSVT.2020.2987141.
[12] Z. Chang, X. Zhang, S. Wang, S. Ma, and W. Gao, “STAM: A SpatioTemporal Attention Based Memory for Video Prediction,” *IEEE Trans. Multimedia*, vol. 25, pp. 2354–2367, 2023, doi: 10.1109/TMM.2022.3146721.
[13] W. Lu, J. Cui, Y. Chang, and L. Zhang, “A Video Prediction Method Based on Optical Flow Estimation and Pixel Generation,” *IEEE Access*, vol. 9, pp. 100395–100406, 2021, doi: 10.1109/ACCESS.2021.3096788.
[14] Y. Zhao and Y. Dou, “Pose-Forecasting Aided Human Video Prediction with Graph Convolutional Networks,” *IEEE Access*, vol. 8, pp. 147256–147264, 2020, doi: 10.1109/ACCESS.2020.2995383.
[15] Y. Wang *et al.*, “PredRNN: A Recurrent Neural Network for Spatiotemporal Predictive Learning,” Apr. 2022, [Online]. Available: http://arxiv.org/abs/2103.09504
[16] Y. Wang, Z. Gao, M. Long, J. Wang, and P. S. Yu, “PredRNN++: Towards A Resolution of the Deep-in-Time Dilemma in Spatiotemporal Predictive Learning,” 2018.
[17] J. Su, W. Byeon, J. Kossaifi, F. Huang, J. Kautz, and A. Anandkumar, “Convolutional Tensor-Train LSTM for Spatio-temporal Learning,” Oct. 2020, [Online]. Available: http://arxiv.org/abs/2002.09131
[18] S. Lee, H. G. Kim, D. H. Choi, H.-I. Kim, and Y. M. Ro, “Video Prediction Recalling Long-term Motion Context via Memory Alignment Learning.” [Online]. Available: https://github.com/sangmin-git/LMC-Memory
[19] Z. Qin and Q. Dai, “A 3D-CNN and multi-loss video prediction architecture,” *Applied Intelligence*, vol. 55, no. 6, Apr. 2025, doi: 10.1007/s10489-025-06328-1.
[20] Z. Hu, T. Turki, and J. T. L. Wang, “Generative Adversarial Networks for Stochastic Video Prediction with Action Control,” *IEEE Access*, vol. 8, pp. 63336–63348, 2020, doi: 10.1109/ACCESS.2020.2982750.
[21] Z. Li and F. Liu, “Scalable video transformer for full-frame video prediction,” *Computer Vision and Image Understanding*, vol. 249, Dec. 2024, doi: 10.1016/j.cviu.2024.104166.
[22] Z. Yang, X. Yang, and Q. Lin, “PTCT: Patches with 3D-Temporal Convolutional Transformer Network for Precipitation Nowcasting,” Jun. 2022, [Online]. Available: http://arxiv.org/abs/2112.01085
[23] X. Ye and G. A. Bilodeau, “VPTR: Efficient Transformers for Video Prediction,” in *Proceedings - International Conference on Pattern Recognition*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 3492–3499. doi: 10.1109/ICPR56361.2022.9956707.
[24] M. Mathai, Y. Liu, and N. Ling, “A Hybrid Transformer-LSTM Model With 3D Separable Convolution for Video Prediction,” *IEEE Access*, vol. 12, pp. 39589–39602, 2024, doi: 10.1109/ACCESS.2024.3375365.
[25] Y. Tang, P. Dong, Z. Tang, X. Chu, and J. Liang, “VMRNN: Integrating Vision Mamba and LSTM for Efficient and Accurate Spatiotemporal Forecasting.” [Online]. Available: https://github.com/yyyujintang/VMRNN-PyTorch.
[26] C. Zhang, J. Tian, L. Gong, Y. Zhang, F. Wu, and H. Pan, “A Video Frame Prediction Method Based on Latent-Space Autoregressive Modeling,” *Applied Sciences*, vol. 16, no. 9, p. 4423, May 2026, doi: 10.3390/app16094423.
[27] Z. Liu *et al.*, “ConvTransformer: A Convolutional Transformer Network for Video Frame

Synthesis," Jun. 2021, [Online]. Available: http://arxiv.org/abs/2011.10185
[28] C. V. Aravinda, T. Al-Shehari, N. A. Alsadhan, S. Shetty, G. Padmajadevi, and K. R. U. K. Reddy, "A novel hybrid architecture for video frame prediction: combining convolutional LSTM and 3D CNN," *J. Real. Time. Image Process.*, vol. 22, no. 1, Feb. 2025, doi: 10.1007/s11554-025-01626-w.
[29] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Long, and L. Fei-Fei, "EIDETIC 3D LSTM: A MODEL FOR VIDEO PREDICTION AND BEYOND."
[30] Q. Huang, Z. Li, L. Zheng, and T. Yang, "Video frame prediction with dual-stream deep network emphasizing motions and content details," *Appl. Soft Comput.*, vol. 125, Aug. 2022, doi: 10.1016/j.asoc.2022.109170.
[31] C. Li and X. Chen, "Future video frame prediction based on generative motion-assistant discriminative network," *Appl. Soft Comput.*, vol. 135, Mar. 2023, doi: 10.1016/j.asoc.2023.110028.
[32] N. Ranjan, S. Bhandari, Y. C. Kim, and H. Kim, "Video Frame Prediction by Joint Optimization of Direct Frame Synthesis and Optical-Flow Estimation," *Computers, Materials and Continua*, vol. 75, no. 2, pp. 2615–2639, 2023, doi: 10.32604/cmc.2023.026086.
[33] H. Huang and Y. Guan, "A Dual-Branch Spatial-Temporal Learning Network for Video Prediction," *IEEE Access*, vol. 12, pp. 73258–73267, 2024, doi: 10.1109/ACCESS.2024.3394209.
[34] J. Lee and G. Kim, "CNN-Based Time Series Decomposition Model for Video Prediction," *IEEE Access*, vol. 12, pp. 131205–131216, 2024, doi: 10.1109/ACCESS.2024.3458460.
[35] J. Lai, L. Gan, J. Zhu, H. Liu, and L. Gao, "Exploring Spatial Frequency Information for Enhanced Video Prediction Quality," *IEEE Trans. Multimedia*, vol. 26, pp. 8955–8968, 2024, doi: 10.1109/TMM.2024.3384062.
[36] B. Singh, D. Singh, R. Kaushal, A. Halder, and P. Chattopadhyay, "GSSTU: Generative Spatial Self-Attention Transformer Unit for Enhanced Video Prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 3, pp. 4625–4638, 2025, doi: 10.1109/TNNLS.2024.3359716.
[37] S. Mokssit, D. Bonilla Licea, B. Guermah, and M. Ghogho, "A Spatial Transformation Based Next Frame Predictor," *IEEE Access*, vol. 13, pp. 5406–5423, 2025, doi: 10.1109/ACCESS.2024.3524997.
[38] F. Naz, L. She, C. Zhang, and J. Shao, "A two-stage trained hybrid Unet-ConvLSTM2D for enhanced precipitation nowcasting," *Environmental Modelling and Software*, vol. 192, Aug. 2025, doi: 10.1016/j.envsoft.2025.106532.
[39] D. Zhou *et al.*, "Taming Teacher Forcing for Masked Autoregressive Video Generation," Institute of Electrical and Electronics Engineers (IEEE), Aug. 2025, pp. 7374–7384. doi: 10.1109/cvpr52734.2025.00691.
[40] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised Learning of Video Representations using LSTMs."
[41] C. Finn, I. G. Openai, S. Levine, and G. Brain, "Unsupervised Learning for Physical Interaction through Video Prediction."
[42] M. Mathai, Y. Liu, and N. Ling, "A Hybrid Transformer-LSTM Model With 3D Separable Convolution for Video Prediction," *IEEE Access*, vol. 12, pp. 39589–39602, 2024, doi: 10.1109/ACCESS.2024.3375365.