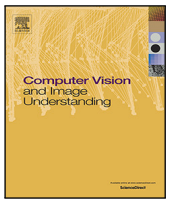
Computer Vision and Image Understanding 259 (2025) 104387

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu



Continuous conditional video synthesis by neural processes

Xi Ye *, Guillaume-Alexandre Bilodeau

LITIV Lab, Polytechnique Montréal, P.O. Box 6079, Station centre-ville, Montreal, H3C3A7, Canada

ARTICLE INFO

Communicated by Feng Liu

Keywords:
Continuous video synthesis
Frame prediction
Frame interpolation
Neural processes
Implicit neural representation
Transformers

ABSTRACT

Different conditional video synthesis tasks, such as frame interpolation and future frame prediction, are typically addressed individually by task-specific models, despite their shared underlying characteristics. Additionally, most conditional video synthesis models are limited to discrete frame generation at specific integer time steps. This paper presents a unified model that tackles both challenges simultaneously. We demonstrate that conditional video synthesis can be formulated as a neural process, where input spatio-temporal coordinates are mapped to target pixel values by conditioning on context spatio-temporal coordinates and pixel values. Our approach leverages a Transformer-based non-autoregressive conditional video synthesis model that takes the implicit neural representation of coordinates and context pixel features as input. Our task-specific models outperform previous methods for future frame prediction and frame interpolation across multiple datasets. Importantly, our model enables temporal continuous video synthesis at arbitrary high frame rates, outperforming the previous state-of-the-art. The source code and video demos for our model are available at https://npvp.github.io.

1. Introduction

Conditional video synthesis for interpolating frames or extrapolating future frames has gained significant attention due to its wide range of applications in areas such as anomaly detection (Hao et al., 2022), autonomous driving, and robotics. In this paper, we focus on two closely related conditional video synthesis tasks, video frame interpolation (VFI) and video future frame prediction (VFP). VFI involves generating intermediate frames between existing frames, while VFP aims at generating future frames based on past frames. VFP is more challenging due to the increased uncertainties in future predictions.

Traditionally, VFI and VFP have been addressed using two distinct approaches. For instance, all VFI methods depend on context information from both the past and future to capture the motion, which is normally represented by optical flow (Niklaus and Liu, 2020) or local convolution kernels (Niklaus et al., 2021). These models cannot be adapted to solve the VFP problem due to the lack of future context. Moreover, VFI methods normally require high frame rate videos and rely on supervised training of neural network models to interpolate missing intermediate frames based on downsampled low frame rate input videos. In contrast, most VFP models employ Convolutional-LSTMs (ConvLSTMs) for autoregressive future frame prediction (Chang et al., 2021), which makes them incapable to perform frame interpolation.

Therefore, this paper aims to unify multiple conditional video synthesis tasks by proposing a novel unsupervised continuous conditional video synthesis model. Our first motivation to unify multiple conditional video synthesis tasks with one model stems from their frequent applications in video processing. For example, they can contribute to improve object detection and data association in object tracking, as missing frames can be rebuilt with VFI or the video frame rate can be improved, and VFP can be used to predict the context changes surrounding an object in a frame to track it more robustly. Another motivation is the effectiveness of multi-task learning as a regularization technique for improved representation learning (Goodfellow et al., 2016). Therefore, we posit that a unified model offers benefits for each individual task.

Additionally, a common limitation in most conditional video synthesis models is generating frames solely at a fixed frame rate, i.e., discrete prediction. This is problematic since the real world exhibits continuity across the spatio-temporal domain. Consequently, we also aim to devise a conditional video synthesis model capable of recovering the inherent continuous signal of real-world data from discrete datasets. This capability opens up valuable possibilities, including generating videos with arbitrary high frame rates and creating climate videos with irregular time intervals (Park et al., 2021).

Therefore, our method leverages neural processes (NPs) (Garnelo et al., 2018) and an implicit neural representation (INR) (Tancik et al., 2020; Sitzmann et al., 2020) to achieve unsupervised continuous conditional video synthesis. While NPs have been successfully applied to

* Corresponding author.
E-mail address: xi.ye@polymtl.ca (X. Ye).

https://doi.org/10.1016/j.cviu.2025.104387
Received 26 March 2024; Received in revised form 3 October 2024; Accepted 8 May 2025
Available online 29 May 2025

image completion (Garnelo et al., 2018; Sitzmann et al., 2020), our work is the first to achieve conditional video synthesis using neural processes. In addition to VFI and VFP, our model can also handle video past frame extrapolation (VPE) and video random missing frame completion (VRC) due to the flexibility of NPs.

To formulate conditional video synthesis as a neural process, we created a supervised mapping from any target spatio-temporal frame coordinates to target pixel values, given observed context coordinates and pixel values. We encode the spatio-temporal coordinates using an implicit neural representation learning model, a Fourier Feature Network (FFN), for continuous synthesis. Specifically, we employ a convolutional neural network (CNN) encoder to extract features from each frame. A Transformer-based prediction model parameterizes the neural process, taking target coordinates representations as inputs and conditioning on context coordinates representations and frame features. The model outputs target frame features, which are then fed into a CNN decoder to reconstruct the frame pixels. We develop both a deterministic predictor and a stochastic predictor, with the latter being utilized for tasks with higher uncertainty, such as VFP and VPE. In summary, our main contributions are:

- We propose the first neural process model for conditional video synthesis, that we named NPVP, which addresses VFP, VFI, VPE, and VRC tasks using with one model. We show that a unified model outperforms task-specific models;
- Our work pioneers the adaptation of implicit neural representation for temporal continuous VFP. Our model enables temporal continuous video synthesis and surpasses the performance of state-of-the-art (SOTA) model in continuous VFP.
- Our model outperforms existing models for VFI and achieves competitive results with SOTA models for VFP.

This paper extends the work in Ye and Bilodeau (2023) by introducing a deterministic variant of the original stochastic model. The deterministic model is better suited for traffic decision-making applications, as it produces a single prediction that generally aligns with the ground-truth. In contrast, the stochastic prediction model generates multiple predictions, requiring additional effort to select the best one for decision making, despite offering a better visual quality. Thus, the deterministic model broadens the applicability of our method. Besides, presenting both the deterministic and stochastic variant of our NPVP model is critical for enhancing our understanding of neural processes-based conditional video synthesis models. We extensively compare the deterministic and stochastic NPVP models to discern their respective advantages and disadvantages. Furthermore, we evaluate the continuous VFP performance of NPVP, a dimension not covered in previous work (Ye and Bilodeau, 2023). Lastly, we meticulously analyze several limitations of our work and propose promising future research to address them.

2. Background

Neural processes (NPs). Garnelo et al. (2018) introduced neural processes (NPs) for modeling the predictive distribution $p(f(T)|C,T)$, where $C = (X_C, Y_C) = \{(x_i, y_i)\}_{i\in\mathcal{I}(C)}$ represents a set of labeled contexts and $T = X_T = \{x_i\}_{i\in\mathcal{I}(T)}$ denotes an unlabeled target set. Here, $\mathcal{I}(S)$ denotes the indices of data points in set S and the function $f : X \to Y$ maps domain X to Y. NPs exhibit efficiency by combining the strengths of Gaussian processes and deep neural networks. They encode and aggregate the contexts C into a fixed-dimensional context embedding, which serves as input along with T to parameterize $p(f(T)|C,T)$ using a neural network. NPs possess the important property of permutation invariance for both C and T (Garnelo et al., 2018). Extensions of NPs include a latent variable version that incorporates the uncertainty of $f(T)$ through a variational autoencoder (VAE). Addressing the underfitting issue of NPs, Kim et al. (2019) proposed an attention mechanism to replace the context feature aggregation operation. NPs are required to exhibit scalability, flexibility, and permutation invariance, leading to successful applications in image completion (Kim et al., 2019; Sitzmann et al., 2020). In this context, x_i represent pixel coordinates, y_i represent pixel values. Leveraging permutation invariance, NPs can predict missing pixel values conditioned on context pixels in arbitrary patterns.

Implicit neural representations (INRs). INRs (Tancik et al., 2020; Sitzmann et al., 2020) address the spectral bias problem in neural networks, enabling a continuous mapping between input coordinates and target signal values (e.g., pixel values). Two main types of INRs exist. The first type is the Fourier Feature Network (FFN) (Tancik et al., 2020), which employs a Fourier feature mapping to effectively capture high-frequency signal components for the input of multiple layer perceptron (MLP). The second type is the SInusoidal REpresentation Network (SIREN) (Sitzmann et al., 2020), utilizing sinusoidal activations to continuously represent signals with fine details. Both FFN and SIREN demonstrate efficiency, and recent studies established their equivalence (Benbarka et al., 2022).

3. Related work

Any video to video synthesis task can be considered as a conditional video synthesis, including video translation between different domains (Wang et al., 2018a), video super-resolution (Song et al., 2022), VFI, and VFP. Our focus is specifically on VFI and VFP-related work. Classical supervised VFI models rely on optical flow (Niklaus and Liu, 2020) or kernel-based methods (Niklaus et al., 2021) to learn motion for intermediate frames. However, these models require high frame rate training datasets, which can be costly to obtain. Recently, there has been the emergence of unsupervised VFI models, such as the cycle consistency-based model proposed by Reda et al. (2019). Another notable approach, VideoINR (Chen et al., 2022), employs optical flow-based CNN models and successfully utilizes implicit neural representation for continuous VFI.

VFP models fall into different categories, including deterministic models (Wu et al., 2020), stochastic models (Babaeizadeh et al., 2018; Denton and Fergus, 2018), pixel-direct generation models (Ye and Bilodeau, 2022), and transformation-based models (Chang et al., 2022). Most VFP models are autoregressive, relying on ConvLSTMs or Transformers. However, recent research has introduced promising non-autoregressive VFP models (Ye and Bilodeau, 2022; Voleti et al., 2022). Vid-ODE (Park et al., 2021) combines ConvLSTMs with a neural ordinary differential equation (ODE) solver, unifying VFP and VFI into a single model capable of generating temporally continuous video frames.

While MCVD (Voleti et al., 2022) extends 3D CNN-based diffusion models for video generation, it differs from an NP model and lacks the ability to perform continuous synthesis. In contrast, our model leverages the flexibility of NPs and can handle video random missing frames completion (VRC), surpassing MCVD in terms of flexibility. Additionally, our model incorporates stochastic prediction using a VAE instead of a diffusion model.

Our model stands out from previous work in two key aspects. Firstly, none of the previous approaches are designed as neural processes. We argue that a NP is a superior choice due to its permutation invariance, unlike ConvLSTMs or 3D-CNNs. Consequently, our model offers greater flexibility in addressing multiple conditional video generation tasks within a single model. Secondly, implicit neural representation, in conjunction with NPs, enables our model to predict frames at any temporal coordinate, including unseen ones during training. Most previous models are limited to predicting frames at a fixed frame rate determined by the training dataset. Although Vid-ODE (Park et al., 2021) overcomes this limitation through ODE integration, our NP-based model outperforms Vid-ODE in both continuous VFP and VFI, as demonstrated in our experiments. Another exception is VideoINR (Chen et al., 2022), which only handles VFI and does not satisfy the properties of NPs.

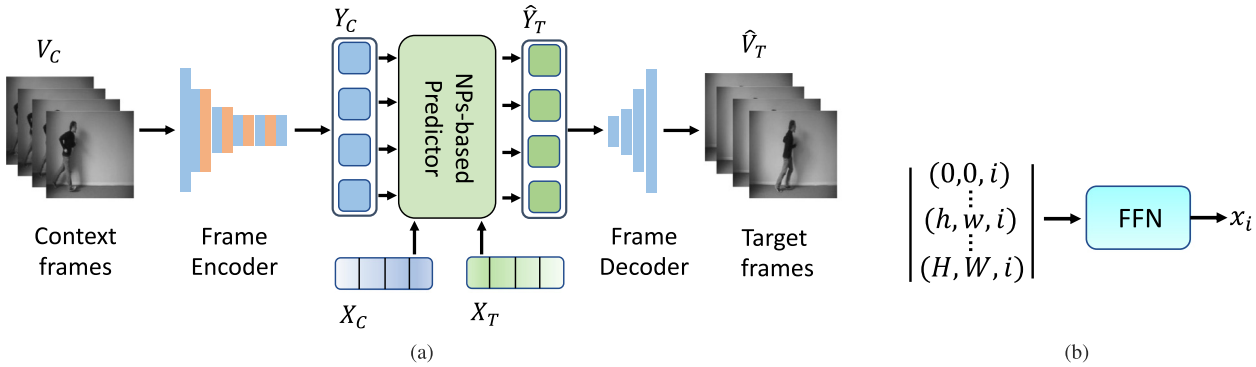


Fig. 1. (a) Overall framework. The blue and orange layers in frame encoder and decoder denotes the convolutional layer and self-attention layer respectively. NPs-based Predictor is implemented as an efficient Transformer. (b) INR. "FFN" denotes the Fourier feature network. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Proposed method

Our proposed overall framework is depicted in Fig. 1(a). Given I_h, I_w, and I_c denoting the image height, width and number of color channels, L_C and L_T denoting the number of context frames and target frames, and H, W, and D denoting the visual feature height, width, and channels, we define the following: $V_C \in \mathbb{R}^{L_C \times I_h \times I_w \times I_c}$, as context frames; $Y_C \in \mathbb{R}^{L_C \times H \times W \times D}$, as the visual features for the context frames; $V_T \in \mathbb{R}^{L_T \times I_h \times I_w \times I_c}$, as the target frames; $Y_T \in \mathbb{R}^{L_T \times H \times W \times D}$, as the visual features for the target frames; $X_C \in \mathbb{R}^{L_C \times H \times W \times D}$ and $X_T \in \mathbb{R}^{L_T \times H \times W \times D}$, as encodings for the context spatio-temporal coordinates and target spatio-temporal coordinates, respectively. To generate target frames $\hat{V}_T$ conditioned on context frames V_C, a conditional video generation model is trained. Initially, a frame encoder extracts the visual features Y_C for the context frames. Subsequently, a predictor utilizes Y_C, X_C, and X_T to predict the target visual feature $\hat{Y}_T$. The encodings X_C and X_T are obtained through a Fourier Feature Network (Fig. 1(b)) and correspond to the context spatio-temporal coordinates and target spatio-temporal coordinates. Finally, the frame decoder reconstructs the target frames $\hat{V}_T$ using $\hat{Y}_T$.

Our model consists of two components: a frame autoencoder and an NPs-based predictor that operates in the feature space. We made this design choice due to the computational expense associated with directly learning a NPs-based model in the pixel space. By operating in the feature space, we can train the model in two stages. Initially, we train the frame autoencoder (encoder and decoder) independently, disregarding the NPs-based predictor. Subsequently, we fix the parameters of the frame autoencoder and train the NPs-based predictor. The detailed architectures of the autoencoder, NPs-based predictors (Figs. 2(a) and 2(b)), and the FFN (Fig. 1(b)) are described in the following sections.

Leveraging the flexibility of NPs, our model allows for the random selection of context frames (V_C) from a video clip, while the remaining frames (V_T) are designated for prediction. This approach enables our model to serve as a general framework for various conditional video generation tasks. For instance, by assigning smaller time coordinates to all context frames compared to the target frames, the model specializes in VFP. However, training the model with randomly selected context frames enables it to address multiple conditional prediction tasks simultaneously.

4.1. Autoencoder

The autoencoder is used to extract the visual features Y_C from the input images and to decode predicted features $\hat{Y}_T$ to generate output images. Our autoencoder is inspired from Pix2Pix (Isola et al., 2017). To enhance the performance, four non-local 2D attention layers (depicted as orange layers in the frame encoder of Fig. 1(a)) from SAGAN (Zhang et al., 2019) are incorporated into the CNN encoder. The frame decoder of Pix2Pix remains unaltered. During training, we employ a L_1 loss between the input frame I and the reconstructed frame $\hat{I}$ using

$$\mathcal{L}_1(I, \hat{I}) = |I - \hat{I}|. \tag{1}$$

The autoencoder is trained independently of the predictor, and the autoencoder parameters remain fixed while training the predictor.

4.2. Fourier feature network for INRs

To obtain the encodings for X_C and X_T, we chose the FFN (Tancik et al., 2020) over SIREN (Sitzmann et al., 2020) due to its ease of implementation and training. The encodings are obtained from spatio-temporal coordinates. The FFN generates implicit neural representations (INRs) X_C and X_T, which encode the spatio-temporal location information of context features Y_C and target features $\hat{Y}_T$.

For a visual feature $y_i \in \mathbb{R}^{H \times W \times D}$ of a single frame, where i denotes the temporal coordinate, the FFN takes the coordinates (h, w, i) of the feature vector for all the spatial locations at i as input. It then generates a D-dimensional coordinate encoding for (h, w, i). This INR is depicted in Fig. 1(b). $x_i \in \mathbb{R}^{H \times W \times D}$ encompasses all the spatial–temporal coordinate encodings for a frame feature y_i. Therefore, X_C and X_T contain all the x_i values for the context and target coordinates, respectively. Specifically, for a 3D input coordinate vector (h, w, i), the FFN initially projects it into a higher-dimensional space using a Gaussian random noise matrix. The resulting projections are then fed into a multi-layer perceptron (MLP) with ReLU activation functions to obtain the output coordinate encoding. The spatio-temporal coordinates are normalized within the range of $[0, 1]$. The FFN and the NPs-based predictor are jointly learned during the training process.

These INRs play a crucial role in training our Transformer-based predictor (see Section 4.3.1), as Transformers are permutation invariant. During test, the INRs exhibit generalization capabilities to unseen input coordinates. This allows us to obtain the coordinate encoding x_i for any real-number temporal coordinate i. By predicting different y_i based on contexts and different target x_i, we achieve continuous generation (see Section 4.3.1). In the case of VFP, if we need to generate target frames beyond the maximum temporal coordinates used during training, we can employ a "block-wise" autoregressive prediction strategy. Specifically, we consider all the generated future (target) frames as the past (context) frames for the subsequent block of future (target) frames. Because our method is block-wise autoregressive, our model achieves faster inference speed in contrast to the standard autoregressive prediction employed in most RNN-based video prediction models, which generates one frame at each inference step.

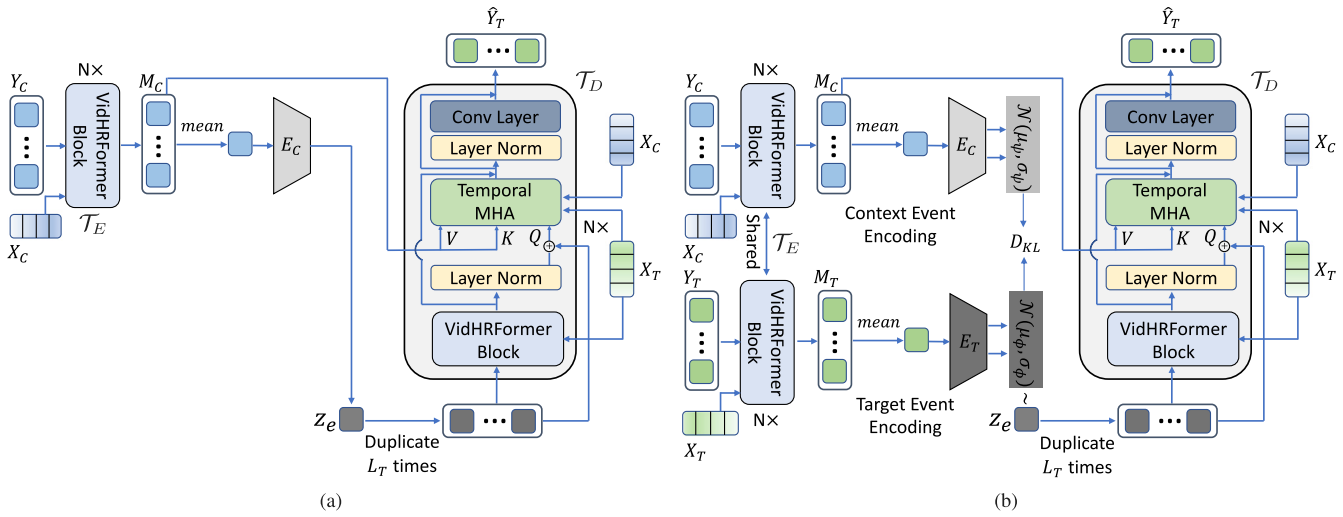


Fig. 2. (a) Deterministic predictor (*NPVP-D*). (b) Stochastic predictor (NPVP-S).

4.3. NPs-based predictor

Given Y_C, X_C, and X_T, the NPs-based predictor predicts the target visual feature $\hat{Y}_T$ of the target frames. We propose two variant for this NPs-based predictor.

4.3.1. Deterministic NPs-based predictor (NPVP-d)

We implemented our deterministic NPs-based predictor using an attentive neural process framework (Kim et al., 2019) built upon VidHRFormer (Ye and Bilodeau, 2022), an efficient spatio-temporal feature learning Transformer. This choice is motivated by the attentive ability of NPs to preserve permutation invariance and overcome underfitting challenges, as well as VidHRFormer suitability for these requirements. Moreover, VidHRFormer has demonstrated successful applications in efficient VFP (Ye and Bilodeau, 2022), and Transformers have shown favorable scaling properties in computer vision tasks (Dosovitskiy et al., 2021).

The objective of a NP is to maximize the conditional log-likelihood, $\log p(Y_T|X_C, Y_C, X_T)$, given contexts (X_C, Y_C) and X_T. Typically, NPs make probabilistic predictions $\hat{Y}_T$ for Y_T, assuming a factorized Gaussian distribution p (Garnelo et al., 2018; Kim et al., 2019). However, for video frame generation, we argue that a simpler point prediction is more suitable. The dimensionality of Y_T is significantly higher compared to the regression datasets or images in previous works (Garnelo et al., 2018; Kim et al., 2019). Consequently, predicting the covariance, even if diagonal (factorized), becomes computationally expensive.

We assume a Laplacian distribution with a constant scale parameter to model p. Consequently, maximizing the log-likelihood is equivalent to minimizing an L_1 loss, specifically $|Y_T - \hat{Y}_T|$. However, we found that solely learning with the L_1 loss in the latent feature space fails to generate predictions with satisfactory visual quality in practice. This limitation arises because the L_1 loss disregards the curvature of the latent feature manifold learned by the frame autoencoder (Shao et al., 2018). To address this issue, we also utilize the fixed frame decoder to reconstruct target frames $\hat{V}_T$ from $\hat{Y}_T$, i.e., concurrently minimizing an additional pixel reconstruction L_1 loss, $|V_T - \hat{V}_T|$. Through this approach, the supervisory signal provided by the pixel L_1 loss minimizes the geodesic distance between Y_T and $\hat{Y}_T$ (Bhagat et al., 2020). Then, the loss function of the *NPVP-D* is

$$\mathcal{L}_{det} = |V_T - \hat{V}_T| + \gamma|Y_T - \hat{Y}_T|, \tag{2}$$

where γ is a hyperparameter. In theory, a vanilla NP is unable to generate coherent video sequences due to the independence of each time step (Garnelo et al., 2018). However, the temporal multihead-attention (MHA) module in the decoder exchanges temporal information, and *NPVP-D* is able to generate video with good temporal consistency as long as Eq. (2) is well optimized.

In detail, the architecture of *NPVP-D* (see Fig. 2(a)) can be formalized by the following operations,

$$M_C = \mathcal{T}_E(X_C, Y_C) \tag{3}$$

$$z_e = E_C(mean(M_C)) \tag{4}$$

$$\hat{Y}_T = \mathcal{T}_D(X_T, X_C, M_C, z_e), \tag{5}$$

where $\mathcal{T}_E : X_C \times Y_C \rightarrow M_C \in \mathbb{R}^{L_C \times H \times W \times D}$ denotes the context Transformer encoder, constructed using multiple VidHRFormer blocks (Ye and Bilodeau, 2022). The VidHRFormer block employs a spatio-temporal separated attention mechanism to ensure permutation invariance along the temporal dimension. $\mathcal{T}_E$ takes X_C as the input positional encodings. We hypothesize that a latent representation called the "event variable", denoted as $z_e \in \mathbb{R}^{H \times W \times D}$, generates all target visual features. To obtain z_e, we first average M_C along the temporal dimension. Then, we pass the result through an event encoder $E_C : \mathbb{R}^{H \times W \times D} \rightarrow \mathbb{R}^{H \times W \times D}$, which is a small CNN. The use of the *mean* operation is motivated by its efficiency as an aggregation method and its permutation invariance.

Finally, the generation of $\hat{Y}_T$ is achieved by conditioning on (X_T, X_C, M_C, z_e) using another Transformer $\mathcal{T}_D$. The architecture of $\mathcal{T}_D$ block aligns with the Transformer decoder block employed in VPTR (Ye and Bilodeau, 2022). Notably, we duplicate the event variable L_T times and input it as the initial query of $\mathcal{T}_D$ for each target frame feature. Additionally, X_T is incorporated into $\mathcal{T}_D$ as positional encodings. Consequently, we can generate $\hat{Y}_T$ with any desired frame rate, allowing for continuous prediction. This is accomplished by providing any desired X_T, which is effortlessly produced by the trained FFN. On the contrary, VPTR (Ye and Bilodeau, 2022) could only predict future frames with a fixed frame rate due the limitation of a fixed number of learned frame queries.

4.3.2. Stochastic NPs-based predictor (NPVP-S)

NPVP-D cannot take into account the randomness and only predicts the average of all possible outcomes (Babaeizadeh et al., 2018). Therefore, a stochastic model is also desirable. For example, a ball with random motion can move toward any direction, even though we will observe only one outcome. The stochastic predictor can generate different random predictions for the same contexts.

In order to achieve coherent sequence sampling, we assume z_e to be randomly sampled from a learned event space, instead of simply being deterministically derived from the contexts, and thus z_e explains the

Table 1
Frame interpolation (VFI) results. p and f denote the number of past frames and number of future frames respectively, k denotes the number of intermediate frames to interpolate. LPIPS is reported in 10^{-3} scale. **Smaller $p+f$ and larger k means a harder VFI task**. †: $p=8$, $f=7$; for our other models, p equals to f. **Boldface**: best results. *Blue*: second best results.

Models	KTH				SM-MNIST				BAIR			
	$(p+f \rightarrow k)$	PSNR↑	SSIM↑	LPIPS↓	$(p+f \rightarrow k)$	PSNR↑	SSIM↑	LPIPS↓	$(p+f \rightarrow k)$	PSNR↑	SSIM↑	LPIPS↓
SVG-LP (Denton and Fergus, 2018)	(*18→7*)	28.13	0.883	–	(*18→7*)	13.54	0.741	–	(*18→7*)	18.65	0.846	–
SDVI full (Xu et al., 2020)	(*18→7*)	29.19	0.901	–	(*18→7*)	16.03	0.842	–	(*18→7*)	21.43	0.880	–
Vid-ODE (Park et al., 2021)	–	31.77	0.911	48	–	–	–	–	–	–	–	–
MCVD (Voleti et al., 2022)	(*15→10*)	34.67	0.943	–	(*10→10*)	20.94	0.854	–	(*4→5*)	*25.16*	*0.932*	–
MCVD (Voleti et al., 2022)	(*10→5*)	35.61	0.963	–	(*10→5*)	27.69	0.940	–	–	–	–	–
NPVP-D (ours)	(*15→10*)†	33.25	0.959	23.76	(10→10)	28.77	0.975	17.41	(*18→7*)	22.29	0.903	24.87
	(10→5)	*36.89*	*0.978*	**6.68**	(10→5)	*34.19*	**0.994**	*4.52*	(*4→5*)	23.22	0.914	22.48
NPVP-S (ours)	(15→10)†	33.60	0.969	22.30	(10→10)	28.11	0.958	17.30	(*18→7*)	22.97	0.909	*21.78*
	(10→5)	**37.17**	**0.984**	*10.5*	(10→5)	**34.34**	*0.992*	**4.10**	(*4→5*)	**25.28**	**0.933**	**14.66**

uncertainty during the prediction of Y_T. Theoretically, we can describe the generative process of Y_T as:

$$p(Y_T|X_C,Y_C,X_T) = \int p(Y_T|X_T,X_C,Y_C,z_e)q(z_e|X_C,Y_C)dz_e, \tag{6}$$

where $q(z_e|X_C,Y_C)$ denotes a conditional prior distribution for z_e. Leveraging the VAE framework, the learning process of *NPVP-S* involves maximizing the evidence lower bound (ELBO):

$$ELBO = \mathbb{E}_{q_\phi(z_e|X_T,Y_T)}[\log p(Y_T|X_T,X_C,Y_C,z_e)] - \beta D_{KL}(q_\phi(z_e|X_T,Y_T) \parallel q_\psi(z_e|X_C,Y_C)), \tag{7}$$

where β is a hyperparameter. The first term of the right-hand side of Eq. (7) can be parameterized the same way as the deterministic predictor, i.e., Eq. (2), which forces the predictor to reconstruct Y_T. We assume that both context frames and target frames originate from the same latent event space. Consequently, the KL divergence of Eq. (7) serves as a regularization term to achieve this, ensuring that the sampled z_e from the targets does not deviate excessively from z_e sampled from the context. During training, z_e is sampled from an approximated posterior event space $\mathcal{N}(\mu_\phi,\sigma_\phi)$ (a factorized Gaussian distribution) generated by the target features and coordinates via the re-parameterization trick,

$$\mu_\phi,\sigma_\phi = E_T(mean(M_T)) = E_T(mean(\mathcal{T}_E(X_T,Y_T))), \tag{8}$$

where E_T is the target event encoder. Eq. (8) formalizes the detailed architecture of the target event encoding path in Fig. 2(b). During test, the ground-truth Y_T is unavailable, we sample z_e from the learned context prior event space $\mathcal{N}(\mu_\psi,\sigma_\psi)$, which is generated by the context event encoding path of Fig. 2(b), i.e.,

$$\mu_\psi,\sigma_\psi = E_C(mean(M_C)) = E_C(mean(\mathcal{T}_E(X_C,Y_C))). \tag{9}$$

5. Experiments

The performances of the proposed deterministic (*NPVP-D*) and stochastic (*NPVP-S*) variants of our model are evaluated on several realistic video datasets, including Cityscapes (Cordts et al., 2016), KITTI (Geiger et al., 2013), KTH (Schuldt et al., 2004), BAIR (Ebert et al., 2017) and a synthetic dataset Stochastic Moving MNIST (SM-MNIST) (Denton and Fergus, 2018). We adopted the experimental configurations used in previous works to report the quantitative results of Fréchet Video Distance (FVD) (Unterthiner et al., 2019), Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018), Structural Similarity Index Measure (SSIM) (Wang et al., 2004) and Peak Signal-to-Noise Ratio (PSNR). For deterministic models, average metrics are reported. In the case of stochastic models, similarly to previous work, we sample 100 different predictions for each test example and report the SSIM, LPIPS, PSNR of the best prediction, and the average FVD among the generated samples.

For fair comparisons, we report the results of all compared methods using the values coming from their original papers unless explicitly stated otherwise. Consequently, any missing metric value for previous works denote that it was not reported. Furthermore, to ensure a fair comparison with previous task-specific models, we initially train separate models for VFP and VFI. This involves following the identical training procedures employed by those task-specific models. Subsequently, we present the results of unified models that are not task-specific.

To reduce the learning burden, we firstly train the self-attention enhanced convolutional autoencoder, which is fixed during the learning of the NPs-based Transformer predictor in the latent space. The autoencoder is optimized using Adam, while the NPs-based predictor uses AdamW, both with a learning rate of $1e-4$ and a cosine learning rate scheduler with warm restarts. For code and checkpoints, please refer to https://github.com/XiYe20/NPVP.

5.1. Evaluation of our task-specific models for frame interpolation (VFI)

We followed the experimental protocol of Voleti et al. (2022). For the VFI task, the model is trained to generate k intermediate frames, i.e., target frames, given p past frames and f future frames as the context.

Table 1 presents the VFI results. Regarding the KTH dataset, both *NPVP-D* (*15→10*) and *NPVP-S* (*15→10*) models outperform the SOTA method MCVD (*15→10*) in terms of SSIM. Additionally, they outperform Vid-ODE across all metrics by a large margin. It is worth noting that a smaller $p+f$ and a larger k correspond to a more challenging VFI task. Therefore, we can conclude that our models outperform SVG-LP and SDVI full as well, because (*15→10*) is harder than (*18→7*). Furthermore, both *NPVP-D* (*10→5*) and *NPVP-S* (*10→5*) outperform MCVD (*10→5*) in terms of both SSIM and PSNR. Generally, *NPVP-S* exhibits better overall performance compared to *NPVP-D*, except for the LPIPS in (*10→5*) frame interpolation task. Besides, Both MCVD and our models perform better in the (*10→5*) VFI than in the (*15→10*) VFI. We attribute this result to the fact that 10 context frames provide sufficient contextual information for the KTH dataset, and interpolating 5 frames is less challenging than interpolating 10 frames.

Our *NPVP-D* (*10→10*) model demonstrates superior performance in terms of both PSNR and SSIM compared to previous methods on the SM-MNIST dataset, even for the easier (*18→7*) and (*10→5*) tasks. This indicates that our model is capable of interpolating more high-quality intermediate frames given the same number of context frames. Furthermore, MCVD is outperformed by *NPVP-D* even if it produces multiple predictions per test example and reports the metrics for the best prediction, whereas our deterministic model only generates a single prediction. As expected, the *NPVP-D* (*10→5*) model significantly outperforms the *NPVP-D* (*10→10*) model. Notably, the improvement of *NPVP-S* over *NPVP-D* is only marginal due to the limited randomness in the intermediate frames, allowing a deterministic model to achieve satisfactory VFI results for the SM-MNIST dataset.

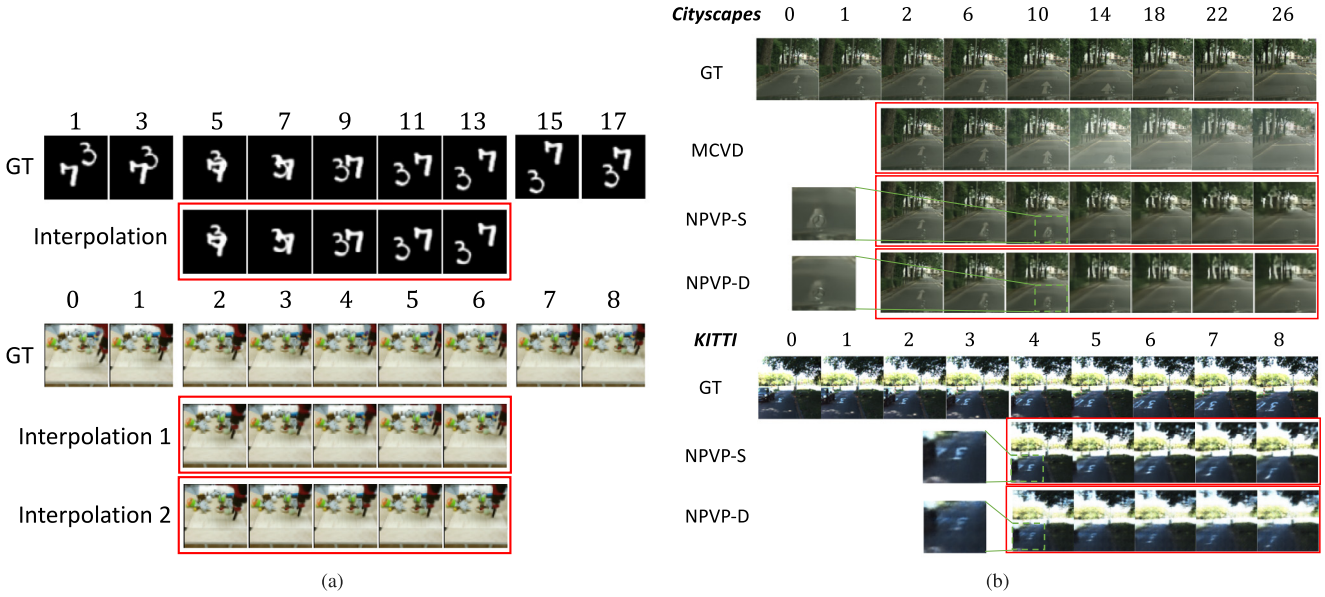


Fig. 3. (a) VFI examples on SM-MNIST by *NPVP-D* and BAIR by *NPVP-S*. For stochastic VFI of BAIR, different robot arm shapes and movements are observed between two interpolations. (b) Future frame prediction (VFP) examples on Cityscapes and KITTI. Frames inside the red boxes are future frames predicted by the model. Zoomed-in regions (dashed green box) demonstrate *NPVP-S* predicts sharper road surface markings. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

For the BAIR dataset, our deterministic *NPVP-D* (*18→7*) model achieves significantly better performance than SVG-LP and SDVI full. However, we observe that there is a performance gap between *NPVP-D* (*4→5*) and MCVD (*4→5*). This is because the robot arm position in each frame of BAIR is independent from other frames, i.e., fully random across the temporal dimension, thus it is difficult to learn the motion of intermediate frames given context frames, unlike KTH, where the natural human motion in missing frames are mostly constrained by the past and future movements. The same applies for SM-MNIST, where the randomness only occurs when the characters bounce off the boundaries. Most of the times, the character trajectories of intermediate frames can be predicted based on the past and future frames. In cases with more randomness, our stochastic *NPVP-S* (*4→5*) model improves the performance in terms of all metrics as expected and it also outperforms MCVD. Compared with *NPVP-D* (*18→7*), a similar performance boost is observed for *NPVP-S* (*18→7*).

Moreover, *NPVP-D* and *NPVP-S* have a significantly smaller number of parameters (120M) compared to the SOTA MCVD (320M), suggesting that our model variants are more efficient. Globally, our frame interpolation (VFI) results show that both our models improve the SOTA, benefiting from the efficiency of neural processes in interpolation. In the cases with more randomness, our stochastic model can improve furthermore because the introduced event latent variable accounts for the stochasticity of frame generation. Visual examples of our deterministic VFI on SM-MNIST and stochastic VFI on BAIR are shown in Fig. 3(a).

5.2. Evaluation of our task-specific models for future frame prediction (VFP)

The VFP experimental results are presented in Tables 2, 4, 3 and 5. In the case of the KTH dataset, the models are trained to predict 10 future frames given 10 past frames. During testing, the performance is evaluated by predicting 20 future frames conditioned on 10 past frames using a block-wise autoregressive inference introduced in Section 4.2. As the results shown in Table 2, our *NPVP-S* achieves the best SSIM. Notably, both *NPVP-D* and *NPVP-S* significantly outperform previous methods in terms of LPIPS.

For the SM-MNIST dataset (Table 3), the models predict 10 future frames given 5 past frames for both training and test. Our *NPVP-D* achieves the best SSIM, but we observe a large performance gap between our model and the SOTA in terms of FVD and LPIPS. By examining some prediction examples, we find that the visual quality of the last few frames quickly degrades and they make the whole video clip look unrealistic, which is exactly what FVD and LPIPS focus on. An unstable training of the *NPVP-S* on SM-MNIST is also observed, similar to the stochastic model on KTH. Thus, the performance of NPVP-S is worse than NPVP-D.

Table 2
Future frame prediction (VFP) results on KTH. $p \rightarrow f$ means p past frames used as context to generate f future frames. LPIPS is reported in 10^{-3} scale. **Boldface**: best results. *Blue*: second best results.

Models	KTH, *10 → 20*		
	PSNR↑	SSIM↑	LPIPS ↓
PredRNN++ (Wang et al., 2018b)	28.47	0.865	228.9
STMFANet (Jin et al., 2020)	**29.85**	0.893	118.1
E3D-LSTM (Wang et al., 2018c)	29.31	0.879	–
Conv-TT-LSTM (Su et al., 2020)	28.36	*0.907*	133.4
Vid-ODE (Park et al., 2021)	28.19	0.878	80.0
VPTR-NAR (Ye and Bilodeau, 2022)	26.96	0.879	86.1
MOSO (Sun et al., 2023)	*29.80*	0.822	83.0
MMVP (Zhong et al., 2023)	27.54	0.906	–
NPVP-D (ours)	27.51	0.906	***65.1***
NPVP-S (ours)	27.66	**0.909**	*66.0*

For the BAIR dataset (Table 3), the models are trained to predict 10 future frames given 2 past frames, but 28 future frames are predicted by block-wise autoregressive inference during test. Our deterministic model reaches a comparable performance with previous work. Similar to the VFI task, *NPVP-S* on BAIR dataset outperforms *NPVP-D* and reaches the second best performance in terms of both SSIM and LPIPS. Different from KTH and SM-MNIST, the training of our stochastic model on BAIR is more stable, which may relate to the value of the hyperparameter β. We observe a low variety among different random samples of VFP, i.e., mode collapsing. We suspect that there are two reasons: 1) it is particularly hard to find a good value for the hyperparameter β, which leads to the mode collapsing and the unstable training. Some other VAE-based stochastic VFP models encounter a similar problem. For example, SV2P (Babaeizadeh et al., 2018) proposed a three stages

Table 3
VFP results on SMMNIST and BAIR. $p \rightarrow f$ means p past frames used as context to generate f future frames. LPIPS is reported in 10^{-3} scale. **Boldface**: best results. *Blue*: second best results.

Models	SM-MNIST, *5 → 10*			BAIR, *2 → 28*		
	FVD↓	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
SVG-LP (Denton and Fergus, 2018)	90.81	0.688	*153.0*	17.72	0.815	60.3
Hier-VRNN (Castrejon et al., 2019)	57.17	0.760	**103.0**	–	0.829	**55.0**
STMFANet (Jin et al., 2020)	–	–	–	**21.02**	**0.844**	93.6
VPTR-NAR (Ye and Bilodeau, 2022)	–	–	–	17.77	0.813	70.0
MCVD-concat (Voleti et al., 2022)	*25.63*	0.786	–	17.70	0.797	–
MCVD-spatin (Voleti et al., 2022)	**23.86**	0.780	–	17.70	0.789	–
MAGVIT (Yu et al., 2023)	–	–	–	19.30	0.787	123.0
NPVP-D (ours)	89.64	**0.868**	221.2	17.47	0.817	65.6
NPVP-S (ours)	95.69	*0.817*	188.7	18.15	*0.842*	*57.43*

Table 4
Future frame prediction (VFP) results on KITTI. $p \rightarrow f$ means p past frames used as context to generate f future frames. LPIPS is reported in 10^{-3} scale. **Boldface**: best results. *Blue*: second best results.

Models	KITTI, *4 → 5*	
	SSIM↑	LPIPS↓
PredNet (Lotter et al., 2017)	0.476	629.5
MCNet (Villegas et al., 2017)	0.555	373.9
Voxel Flow (Liu et al., 2017)	0.426	415.9
FVS (Wu et al., 2020)	0.608	304.9
SADM (Bei et al., 2021)	*0.647*	311.6
OPT (Wu et al., 2022)	0.611	**263.5**
NPVP-D (ours)	0.633	297.8
NPVP-S (ours)	**0.661**	*279.0*

Table 5
Future frame prediction (VFP) results on Cityscapes. $p \rightarrow f$ means p past frames used as context to generate f future frames. LPIPS is reported in 10^{-3} scale. **Boldface**: best results. *Blue*: second best results.

Models	Cityscapes, *2 → 28*		
	FVD↓	SSIM↑	LPIPS↓
SVG-LP (Denton and Fergus, 2018)	1300.26	0.574	549.0
VRNN 1L (Castrejon et al., 2019)	682.08	0.609	304.0
Hier-VRNN (Castrejon et al., 2019)	567.51	0.628	264.0
GHVAEs (Wu et al., 2021)	*418.00*	*0.740*	194.0
MCVD-concat (Voleti et al., 2022)	**141.31**	0.690	**112.0**
NPVP-D (ours)	889.12	0.664	234.1
NPVP-S (ours)	768.04	**0.744**	*183.2*

training strategy to achieve stable optimization of VAE, and 2) we only take one latent variable, i.e., a time-invariant latent variable, to account for the stochasticity of whole video sequence, which is limited by the framework of the stochastic NPs. Integrating a time-variant latent variable or hierarchical latent variable could be a potential solution.

For KITTI (Table 4), all models are trained to predict 5 future frames given 4 past frames. Compared with previous methods, our *NPVP-S* reaches the best performance in terms of SSIM and the second best LPIPS. Qualitative results (Fig. 3(b)) show that *NPVP-S* predicts future frames with good visual quality despite the large motion in KITTI dataset, which has a low frame rate of 10 fps. The results on KITTI dataset demonstrate that both *NPVP-S* and *NPVP-D* are capable of challenging real-world traffic future frame prediction.

For the Cityscapes dataset (Table 5), our *NPVP-S* and *NPVP-D* models are trained to predict 10 future frames given 2 past frames, but we predict 28 future frames using block-wise autoregressive inference. Among the methods evaluated, *NPVP-S* achieves the highest SSIM and the second-best LPIPS. The performance gap between our two *NPVP* models and MCVD-concat in terms of FVD can be attributed to the limitation of our chosen VAE that is not expressive enough (Castrejon et al., 2019). It is worth noting that all methods for Cityscapes, except for the MCVD-concat, are VAE-based. MCVD-concat utilizes a denoising diffusion model (DDM) (Ho et al., 2020), and exhibits superior performance in terms of FVD, despite its brightness change issues (Voleti et al., 2022) (see Fig. 3(b)). Previous research by Castrejon et al. (2019) has demonstrated that increasing the levels of latent variables in VAE-based models improves their expressiveness. Meanwhile, DDM can be viewed as a form of hierarchical VAEs with a high number of latent variables, all having the same dimensionality as the video data. Consequently, MCVD-concat achieves the best FVD score due to the heightened expressiveness of its latent variables. Our proposed *NPVP* models may suffer from weaker temporal coherence due to its non-autoregressive prediction. All VAE-based models in Table 5 are autoregressive models. The loss function of our *NPVP* models assumes independence between frames at different time steps to maintain the flexibility of the unified model, despite the exchange of temporal attention information. Nonetheless, *NPVP-S* attains a comparable or superior performance compared to the SOTA methods. As expected, *NPVP-S* outperforms *NPVP-D* by effectively addressing the stochasticity of future prediction. Visual examples of VFP on Cityscapes (Fig. 3(b)) further demonstrate that *NPVP-S* provides superior predictions compared to MCVD (Voleti et al., 2022), which is hindered by brightness change issues.

In general, results show that both our models improve the SOTA in several cases, demonstrating that they are capable of achieving high-quality VFP.

5.3. A unified model for VFI, VFP, VPE and VRC

We trained a unified *NPVP-S* on the KTH dataset to demonstrate that our model is flexible enough to perform VFI, VFP, video past frame extrapolation (VPE), and video random missing frames completion (VRC) with one single model. Moreover, our model enables continuous prediction, i.e., solving all these tasks with an arbitrary high frame rate. To achieve this, we employed a training approach with random contexts. Specifically, given a video clip with a total of L frames, we randomly selected L_C frames as context frames, while the remaining $L_T = L - L_C$ frames served as target frames. The corresponding spatio-temporal coordinates were included for both the context and target frames.

5.3.1. Video synthesis performance with the unified model

Results of our unified model on KTH for four distinct conditional video synthesis tasks are presented in Fig. 4. For the random context sampling, L was set to 20, and the value of L_C varied within the range of 4 to 16. The top row depicts the ground-truth (GT) frames. The red box highlights the target frames generated by the model given the other context frames. Notably, the visual quality of VRC and VFI target frames surpasses that of VPE and VFP. This is because VPE and VFP rely solely on past or future frames, resulting in increased uncertainties and difficulties in predicting target frames. Conversely, VRC and VFI

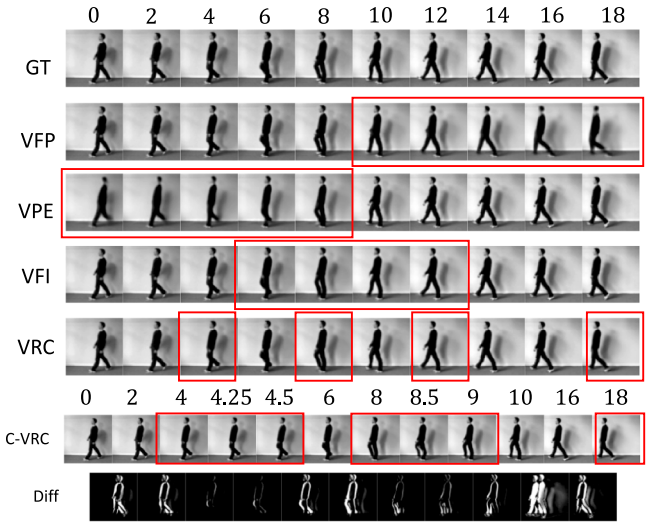


Fig. 4. One model for all tasks. Frames inside the red boxes are target frames generated by the model. C-VRC denotes continuous VRC. Diff are the difference images between neighboring frames of C-VRC to show that they are all different and that the temporal coordinates are taken into account.

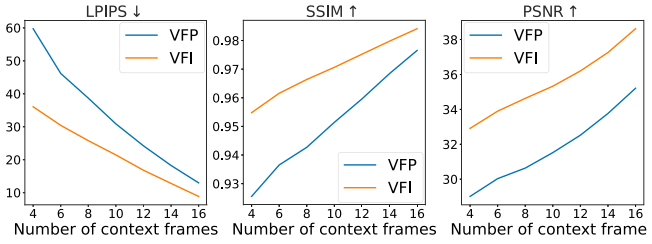


Fig. 5. Metric curves of VFP and VFI on KTH for an increasing number of context frames. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

benefit from scattered context frames across the temporal dimension, providing more accurate motion information. It is worth mentioning that the model trained with random contexts requires more epochs to achieve comparable performance on VFP and VPE tasks, as it tends to minimize loss quickly on easier tasks.

We also examined the quantitative impact of the number of context frames on the visual quality of target frames in VFP and VFI using our unified model (Fig. 5). The results demonstrate a monotonic improvement in all quality metrics as more context frames are provided to the model, i.e., increased context leads to more accurate target frame generation. This aligns with the inherent property of NPs (Garnelo et al., 2018). Additionally, Fig. 5 illustrates the performance difference between VFP and VFI, indicating that VFI is the easier task.

In Table 6, we give a comparison of the VFI and VFP performances between the task-specific *NPVP-S* model and the unified *NPVP-S* model on the KTH and KITTI datasets. For the random context sampling of the unified model on KITTI, the number of context frames L_C varied within the range of 3 to 6, and L was set to 9. Results from the table confirm that the unified model outperforms task-specific models across almost all metrics for both tasks, except for the SSIM of VFP on the KITTI dataset. In short, the results validate our motivation that multi-task learning is advantageous and that the flexibility of our model allows it to capitalize on this type of training.

5.3.2. Continuous prediction performance with the unified model

Finally, our unified model is also able to conduct continuous prediction, as shown at the bottom of Fig. 4. We performed a continuous video random missing frames completion (C-VRC) experiment to showcase the model ability to generate frames at unseen temporal coordinates (e.g., $4.25, 4.5, 8.5$). This highlights our model ability for conducting conditional video synthesis at arbitrary high frame rates.

To compare with the previous continuous prediction model, Vid-ODE (Park et al., 2021), we downsampled the BAIR and KITTI datasets to a 0.5 frame rate during training and evaluated the performance of our model and of Vid-ODE at a 2× frame rate during testing, using the ground-truth high-frame rate test videos for metric calculations. The quantitative results for continuous video frame prediction (VFP) are presented in Table 7. Our *NPVP-S* outperforms Vid-ODE significantly in terms of all metrics, because Vid-ODE is deterministic and has limited capacity. Notably, Vid-ODE performance on the original BAIR dataset is reasonable, but downsampled frame rate videos have a larger temporal gap and Vid-ODE struggles with the increased difficulty caused by this downsampling. Vid-ODE fails to achieve satisfactory results on the large, realistic, and high-resolution KITTI dataset mainly due to its limited model size. Visual examples from the BAIR dataset reveal that Vid-ODE struggles to predict the stochastic motion of the robot arm and experiences rapid degradation in image quality due to accumulated error.

Upon analyzing the generated videos, we observe that the unified model continuous predictions exhibit superior temporal consistency compared to task-specific models. This can be attributed to the random contexts aiding the model in learning more complex conditional distributions (Kim et al., 2019), thereby enhancing the generalization ability of implicit neural representations (INRs). For video examples of different experiments, please visit *https://npvp.github.io*.

5.4. Running time comparison

To demonstrate the efficiency of our proposed NPVP models, we compare the number of parameters and average inference time across various models in Table 8. We selected classical models from three types, a diffusion-based model (MCVD), a recurrent VAE-based model (Hier-VRNN), and a ConvLSTM-based model (Conv-TT-LSTM), as baselines. While Conv-TT-LSTM has the fewest parameters, its inference speed is hindered by the time-consuming vanilla autoregressive prediction. In contrast, our NPVP models offer a balance between model size and the fastest inference speed, benefiting from the efficient block-wise autoregressive prediction, and making them suitable for fast traffic decision-making applications. The MCVD model exhibits a much slower inference due to its iterative reverse diffusion process.

5.5. Ablation study

Table 9 reports the results of our ablation study. The base model (Model 1) used in this study is a deterministic model with a 6-layers of $\mathcal{T}_D$ ($\mathcal{T}_D$-6L). It is trained solely using an L_1 loss in the feature space, without incorporating INR, i.e., the model directly receives spatio-temporal coordinates as input. To investigate the impact of the different components of our models, we gradually modify the architecture and loss function. All models are trained with random contexts, allowing us to evaluate their performance in both VFI and VFP tasks. Specifically, for VFI, we utilize 5 past frames and 5 future frames as input to predict 10 intermediate frames. In the case of VFP, we employ 10 past frames as input to predict 10 future frames.

INR. The inclusion of INR yields a significant improvement in the performance of model 2 in terms of all metrics. This validates the effectiveness of INR. Additionally, qualitative analysis reveals that predictions generated by model 1 lack temporal consistency and exhibit unnatural, choppy motion. We attribute the performance enhancement brought about by INR to three factors: 1) the learned Fourier features provide superior representation of high-frequency details, thereby enhancing visual quality, 2) INR enables the model to learn a continuous mapping from coordinates to video pixels, resulting in improved

Table 6
Comparison of Unified model and task-specific *NPVP-S*. LPIPS is reported in 10^{-3} scale. **Boldface**: best results.

Models	KTH						KITTI			
	$5+5 \rightarrow 10$, VFI			$10 \rightarrow 10$, VFP			$2+2 \rightarrow 5$, VFI		$4 \rightarrow 5$, VFP	
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	SSIM↑	LPIPS↓	SSIM↑	LPIPS↓
Task-specific	33.45	0.960	23.84	29.85	0.924	44.52	0.683	157.10	**0.661**	279.0
Unified	**35.33**	**0.971**	**21.40**	**31.52**	**0.951**	**30.90**	**0.729**	**146.56**	0.650	**216.0**

Table 7
Continuous VFP results on BAIR and KITTI. LPIPS is reported in 10^{-3} scale. **Boldface**: best results. †: We trained the model utilizing the officially released code.

Models	BAIR, $2 \times fps$				KITTI, $2 \times fps$			
	$(p \rightarrow f)$	FVD↓	SSIM↑	LPIPS↓	$(p \rightarrow f)$	FVD↓	SSIM↑	LPIPS↓
Vid-ODE† (Park et al., 2021)	$(2 \rightarrow 28)$	*2948.82*	0.310	322.58	$(4 \rightarrow 5)$	615.98	0.23	591.54
NPVP-S (unified)	$(2 \rightarrow 28)$	**1159.14**	**0.795**	**58.71**	$(4 \rightarrow 5)$	**248.82**	**0.56**	**313.94**

Table 8
Comparison of number of parameters and inference time. †: All models are evaluated with the officially released code on a same Nvidia L40S GPU. **Boldface**: best results. *Blue*: second best results.

Models	KTH, $10 \rightarrow 20$	
	#Params	Inference time (ms)
MCVD† (Voleti et al., 2022)	328.60M	9120.4
Hier-VRNN† (Castrejon et al., 2019)	260.68M	392.2
Conv-TT-LSTM† (Su et al., 2020)	**2.69M**	68.8
NPVP-D (ours)	102.27M	**54.2**
NPVP-S (ours)	103.91M	57.6

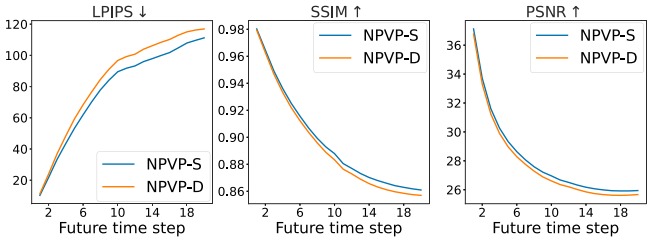


Fig. 6. VFP metric curves of NPVP-D and NPVP-S on KTH for increasing future prediction steps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

continuous prediction, and 3) the learnable Fourier features serve as better multi-dimensional positional encodings, capturing more complex relationships (Li et al., 2021).

Pixel L_1 loss. The incorporation of the pixel L_1 loss leads to performance improvements across nearly all metrics. Notably, model 3 demonstrates a substantial advantage over model 2 in terms of LPIPS. This observation exhibits the positive impact of the pixel L_1 loss on the visual quality of the target frames.

Size of $\mathcal{T}_D$. To examine the impact of $\mathcal{T}_D$ size, the number of layers in $\mathcal{T}_D$ is increased from 6 to 8 ($\mathcal{T}_D$-8L). Comparing the results of model 4 with model 3, we observed improved performance across all VFI and VFP metrics, demonstrating the usefulness of a larger $\mathcal{T}_D$. Notably, LPIPS exhibited a more substantial improvement compared to SSIM and PSNR.

Context Transformer encoder $\mathcal{T}_E$. In Models 1–4, there is no explicit temporal relationship modeling for context frame features. Since effective aggregation of context information is crucial for enhancing NP performance, we introduce the context Transformer $\mathcal{T}_E$ to model the temporal relationships of Y_C and generate M_C for $\mathcal{T}_D$ and z_e, which results in the *NPVP-D* model. Comparing *NPVP-D* with the previous models, we observe significant improvements across all VFP and VFI metrics, particularly for LPIPS.

Stochastic vs Deterministic. Finally, *NPVP-D* is modified to incorporate the VAE architecture, resulting in the stochastic *NPVP-S*. As expected, *NPVP-S* outperforms *NPVP-D* in terms of all metrics for both VFI and VFP. Instead of only predicting the average of all possible outcomes as its deterministic counterpart (Babaeizadeh et al., 2018), *NPVP-S* considers the randomness of prediction by incorporating the event variable. Even though KTH has relatively strong motion regularity and certainty, the speed of arm and leg movements still have large variations, especially for VFP. Consequently, NPVP-S produces frames of higher visual quality, e.g., sharper frames. Importantly, its stochastic generation capability enables NPVP-S to sample predictions that better align with the ground-truth. However, the introduction of the VAE leads to training instability and poorer convergence. To address this, we propose a two-stage training strategy. In the initial stage, we train a deterministic model without considering the target event encoding path and the D_{KL}. Subsequently, we include the target event encoding path and D_{KL} for the training of *NPVP-S*.

We also investigated the quality degradation issue in both *NPVP-D* and *NPVP-S* models for the VFP task. Our analysis, illustrated in Fig. 6, reveals a consistent decline in all three predicted frame metrics with increasing future time steps. This degradation is a pervasive challenge in VFP models due to the accumulation of prediction errors over time. A VFP model with enhanced long-term temporal information modeling could partially mitigate this problem. Nevertheless, Fig. 6 highlights *NPVP-S* outperforming *NPVP-D*, with the performance gap widening as future time steps increase.

6. Limitations

First, the stochastic generation ability of *NPVP-S* is currently limited by the expressiveness of the vanilla VAE, resulting in low diversity. To overcome this limitation, one promising approach is integrating hierarchical VAEs or diffusion models into our neural processed-based video synthesis framework. This integration would enhance the capacity for stochasticity modeling, potentially leading to greater diversity in generated outputs. Additionally, introducing local latent variables for each target timestep, instead of solely relying on a single global latent variable, could further mitigate the issue of limited diversity.

Secondly, our current model is limited to temporal continuous synthesis, and it lacks the capability to synthesize videos with arbitrary resolutions, i.e., spatial continuous synthesis. To address this limitation, a possible solution would be to extend the domain of the function for neural processes from 1D temporal space to the entire 3D spatio-temporal space. This involves predicting latent variables z for each feature at different spatio-temporal locations, rather than predicting event variable z_e for an entire frame at once.

Table 9
Ablation Study on KTH dataset, trained with random contexts. FL_1 denotes feature space L_1 loss. PL_1 denotes pixel space L_1 loss. *NPVP* is the stochastic counterpart of model 5. LPIPS is reported in 10^{-3} scale. **Boldface**: best results. *Blue*: second best results.

	Models							VFI, *5+5 →10*			VFP, *10 → 10*		
	INR	$\mathcal{T}_D$-6L	$\mathcal{T}_D$-8L	$\mathcal{T}_E$	FL_1	PL_1	D_{KL}	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
1		✓			✓			30.58	0.937	50.07	28.05	0.904	71.75
2	✓	✓			✓			31.34	0.953	37.03	28.83	0.920	71.04
3	✓	✓			✓	✓		33.17	0.958	31.11	29.60	0.920	62.96
4	✓		✓		✓	✓		33.20	0.959	29.64	29.85	0.922	57.49
NPVP-D	✓		✓	✓	✓	✓		*33.77*	*0.962*	*26.83*	*30.11*	*0.927*	*53.89*
NPVP-S	✓		✓	✓	✓	✓	✓	**34.07**	**0.972**	**25.59**	**30.37**	**0.941**	**52.18**

7. Conclusion

We introduce a novel continuous conditional video synthesis model that combines neural processes and implicit neural representation. By training with random contexts, our model can address multiple conditional video synthesis tasks, such as video future frame prediction, video frame interpolation, video past frame extrapolation, and video random missing frame completion simultaneously. Moreover, our unified model outperforms the task-specific variants of our model, showing the benefit of multi-task learning. Notably, our model enables predictions at an arbitrary high frame rate. Experimental results demonstrate significant performance improvements over the previous approach in continuous prediction. Additionally, we achieve state-of-the-art performance in video frame interpolation across multiple datasets and comparable results to state-of-the-art models in video future frame prediction.

In future work, we plan to integrate hierarchical VAE or diffusion models into our framework and explore the extension of neural processes to 3D spatio-temporal space, enabling more versatile and advanced video synthesis capabilities.

CRediT authorship contribution statement

Xi Ye: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Guillaume-Alexandre Bilodeau:** Writing – review & editing, Validation, Supervision, Resources, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [RGPIN-2020-04633] and FRQ-NT REPARTI strategic cluster.

Data availability

The datasets used in the paper are public available, and the code are open sourced.

References

Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R., Levine, S., 2018. Stochastic variational video prediction. In: ICLR.
Bei, X., Yang, Y., Soatto, S., 2021. Learning semantic-aware dynamics for video prediction. In: CVPR.
Benbarka, N., Höfer, T., Riaz, H.u.M., Zell, A., 2022. Seeing implicit neural representations as Fourier series. In: WACV.
Bhagat, S., Uppal, S., Yin, Z., Lim, N., 2020. Disentangling multiple features in video sequences using Gaussian processes in variational autoencoders. In: ECCV. pp. 102–117.
Castrejon, L., Ballas, N., Courville, A., 2019. Improved conditional VRNNs for video prediction. In: ICCV.
Chang, Z., Zhang, X., Wang, S., Ma, S., Gao, W., 2022. STRPM: A spatiotemporal residual predictive model for high-resolution video prediction. In: CVPR. pp. 13946–13955.
Chang, Z., Zhang, X., Wang, S., Ma, S., Ye, Y., Xiang, X., Gao, W., 2021. MAU: A motion-aware unit for video prediction and beyond. In: NeurIPS.
Chen, Z., Chen, Y., Liu, J., Xu, X., Goel, V., Wang, Z., Shi, H., Wang, X., 2022. VideoINR: Learning video implicit neural representation for continuous space-time super-resolution. In: CVPR.
Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. pp. 3213–3223.
Denton, E., Fergus, R., 2018. Stochastic video generation with a learned prior. In: ICML.
Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR.
Ebert, F., Finn, C., Lee, A.X., Levine, S., 2017. Self-supervised visual planning with temporal skip connections. In: CoRL.
Garnelo, M., Rosenbaum, D., Maddison, C., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y.W., Rezende, D., Eslami, S.M.A., 2018. Conditional neural processes. In: ICML.
Geiger, A., Lenz, P., Stiller, C., Urtasun, R., 2013. Vision meets robotics: The KITTI dataset. Int. J. Robot. Res. 32 (11), 1231–1237.
Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.
Hao, Y., Li, J., Wang, N., Wang, X., Gao, X., 2022. Spatiotemporal consistency-enhanced network for video anomaly detection. Pattern Recognit. 121, 108232.
Ho, J., Jain, A., Abbeel, P., 2020. Denoising diffusion probabilistic models. In: NeurIPS. vol. 33, pp. 6840–6851.
Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In: CVPR.
Jin, B., Hu, Y., Tang, Q., Niu, J., Shi, Z., Han, Y., Li, X., 2020. Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction. In: CVPR.
Kim, H., Mnih, A., Schwarz, J., Garnelo, M., Eslami, A., Rosenbaum, D., Vinyals, O., Teh, Y.W., 2019. Attentive neural processes. In: ICLR.
Li, Y., Si, S., Li, G., Hsieh, C.J., Bengio, S., 2021. Learnable fourier features for multi-dimensional spatial positional encoding. In: NeurIPS. vol. 34.
Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A., 2017. Video frame synthesis using deep voxel flow. In: International Conference on Computer Vision (ICCV). pp. 4473–4481.
Lotter, W., Kreiman, G., Cox, D., 2017. Deep predictive coding networks for video prediction and unsupervised learning. In: ICLR. pp. 1–18.
Niklaus, S., Liu, F., 2020. Softmax splatting for video frame interpolation. In: CVPR. pp. 5436–5445.
Niklaus, S., Mai, L., Wang, O., 2021. Revisiting adaptive convolutions for video frame interpolation. In: WACV. pp. 1099–1109.
Park, S., Kim, K., Lee, J., Choo, J., Lee, J., Kim, S., Choi, E., 2021. Vid-ODE: Continuous-time video generation with neural ordinary differential equation. In: AAAI.
Reda, F.A., Sun, D., Dundar, A., Shoeybi, M., Liu, G., Shih, K.J., Tao, A., Kautz, J., Catanzaro, B., 2019. Unsupervised video interpolation using cycle consistency. In: ICCV. pp. 892–900.
Schuldt, C., Laptev, I., Caputo, B., 2004. Recognizing human actions: a local SVM approach. In: ICPR.
Shao, H., Kumar, A., Thomas Fletcher, P., 2018. The Riemannian geometry of deep generative models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 315–323.
Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G., 2020. Implicit neural representations with periodic activation functions. In: NeurIPS.

Song, H., Jin, Y., Cheng, Y., Liu, B., Liu, D., Liu, Q., 2022. Learning interlaced sparse Sinkhorn matching network for video super-resolution. Pattern Recognit. 124, 108475.

Su, J., Byeon, W., Kossaifi, J., Huang, F., Kautz, J., Anandkumar, A., 2020. Convolutional tensor-train LSTM for spatio-temporal learning. In: NeurIPS.

Sun, M., Wang, W., Zhu, X., Liu, J., 2023. MOSO: Decomposing MOtion, scene and object for video prediction. pp. 18727–18737.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R., 2020. Fourier features let networks learn high frequency functions in low dimensional domains. In: NeurIPS.

Unterthiner, T., Steenkiste, S.v., Kurach, K., Marinier, R., Michalski, M., Gelly, S., 2019. FVD: A new metric for video generation. In: ICLR Workshop.

Villegas, R., Yang, J., Hong, S., Lin, X., Lee, H., 2017. Decomposing motion and content for natural video sequence prediction. In: ICLR.

Voleti, V., Jolicoeur-Martineau, A., Pal, C., 2022. Masked conditional video diffusion for prediction, generation, and interpolation. In: Advances in Neural Information Processing Systems.

Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 13 (4), 600–612.

Wang, Y., Gao, Z., Long, M., Wang, J., Yu, P.S., 2018b. PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In: ICML.

Wang, Y., Jiang, L., Yang, M.H., Li, L.J., Long, M., Fei Fei, L., 2018c. Eidetic 3D LSTM: A model for video prediction and beyond. In: ICLR.

Wang, T.C., Liu, M.Y., Zhu, J.Y., Liu, G., Tao, A., Kautz, J., Catanzaro, B., 2018a. Video-to-video synthesis. In: NeurIPS. 31.

Wu, Y., Gao, R., Park, J., Chen, Q., 2020. Future video synthesis with object motion prediction. In: CVPR.

Wu, B., Nair, S., Martin-Martin, R., Fei-Fei, L., Finn, C., 2021. Greedy hierarchical variational autoencoders for large-scale video prediction. In: CVPR. pp. 2318–2328.

Wu, Y., Wen, Q., Chen, Q., 2022. Optimizing video prediction via video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17814–17823.

Xu, Q., Zhang, H., Wang, W., Belhumeur, P., Neumann, U., 2020. Stochastic dynamics for video infilling. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2714–2723.

Ye, X., Bilodeau, G.A., 2022. VPTR: Efficient transformers for video prediction. In: ICPR.

Ye, X., Bilodeau, G.A., 2023. A unified model for continuous conditional video prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 3603–3612.

Yu, L., Cheng, Y., Sohn, K., Lezama, J., Zhang, H., Chang, H., Hauptmann, A.G., Yang, M.H., Hao, Y., Essa, I., Jiang, L., 2023. MAGVIT: Masked generative video transformer. pp. 10459–10469.

Zhang, H., Goodfellow, I., Metaxas, D., Odena, A., 2019. Self-attention generative adversarial networks. In: ICML.

Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O., 2018. The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595.

Zhong, Y., Liang, L., Zharkov, I., Neumann, U., 2023. MMVP: Motion-matrix-based video prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4273–4283.